KEKER & VAN NEST, LLP
MICHAEL H. PAGE - #154913
ASHOK RAMANI - #200020
KHARI J. TILLERY - #215669
710 Sansome Street
San Francisco, CA 94111-1704
Telephone: (415) 391-5400
Facsimile: (415) 397-7188

Attorneys for Third-Party Google Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

VISTO CORPORATION,

Plaintiff,

v.

RESEARCH IN MOTION LIMITED, et al.,

Defendant.

Misc. Civil Case No. 3:08-mc-80031-JSW (JL)

Court of Original Jurisdiction
U.S. Dist. Ct.
E. District Texas
Marshall Division
Case No. 2-06-CV-181 TJW

DECLARATION OF KHARI J. TILLERY IN SUPPORT OF THIRD PARTY GOOGLE'S REPLY IN SUPPORT OF MOTION TO QUASH SUBPOENA, OR IN THE ALTERNATIVE, FOR PROTECTIVE ORDER, AND OPPOSITION TO MOTION TO COMPEL

Date: May 7, 2008
Time: 9:30 a.m.
Judge: Magistrate Judge Larson

I, KHARI J. TILLERY, declare and state as follows:

1. I am an attorney licensed to practice law in the State of California and am an associate with Keker & Van Nest, LLP, counsel for third party Google Inc. I make the following statements of my personal knowledge, have knowledge of the facts set forth herein and, if called as a witness, could and would testify competently thereto.

2. Attached hereto as **Exhibit A** is a true and correct copy of an email from Steve Pollinger to me, dated April 1, 2008, with Visto Corporation's proposed declaration attached thereto.

3. Attached hereto as **Exhibit B** is a true and correct copy of a letter from me to Geoffrey L. Smith, dated March 4, 2008.

4. Attached hereto as **Exhibit C** is a true and correct copy of the complaint in *Visto Corporation v.* Research In Motion Limited and Research In Motion Corporation, Eastern District of California Case No. 2-06-CV-181-TJW.

I declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct, and that this declaration was executed on April 23, 2008, at San Francisco, California.

/s/ *Khari J. Tillery*
KHARI J. TILLERY

EXHIBIT 'A'

Khari Tillery

From: spollinger@McKoolSmith.com
Sent: Tuesday, April 01, 2008 7:24 AM
To: Khari Tillery
Cc: gsmith@McKoolSmith.com; mrobson@McKoolSmith.com
Subject: Visto/RIM - first draft of Google Declaration
Attachments: Google Declaration.DOC

Khari,

Per our telephone conversation yesterday, attached is the draft declaration of Google. We look forward to receiving Google's revisions to the declaration by Wednesday of this week.

<<Google Declaration.DOC>>

Steve Pollinger
McKool Smith, P.C.
Suite 1700
300 West 6th Street
Austin, Texas 78701
Direct (512) 692-8702
Fax (512) 692-8744
spollinger@mckoolsmith.com
www.mckoolsmith.com

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VISTO CORPORATION,

Plaintiff and Counterclaim-Defendant,

vs.

RESEARCH IN MOTION LIMITED and RESEARCH IN MOTION CORPORATION,

Defendants and Counterclaim-Plaintiffs.

Civil Action No. 2:06-CV-181 (TJW)

JURY TRIAL DEMANDED

DECLARATION OF GOOGLE INC.

NOW COMES, Google Inc. ("Google") and makes the following Declaration for admission into evidence in the above-identified case. Google has personal knowledge of the matters set forth herein and knows them to be true and correct.

1. The information set forth below is "CONFIDENTIAL-ATTORNEYS' EYES ONLY" pursuant to Paragraphs 3.3. and 4.3 of the Protective Order entered by United States District Court Judge T. John Ward on April 2, 2007, in the above-captioned case.

2. Google e-mail accounts (known as Gmail) are serviced by Google e-mail servers located in the United States and positioned within at least one firewall. All of Google e-mail servers, including its POP, SMTP, and IMAP and e-mail notification servers are located in the United States and are positioned within at least one firewall at their respective locations.

3. Google email accounts (Gmail accounts) can be synchronized with BlackBerry enabled handhelds through systems and services provided by Research in Motion ("RIM"). The only protocols used for performing such synchronization are POP3/SSL, SMTP/SSL,

IMAP/SSL, and Google's HTTP/SSL based custom subscription and notification protocol. All network communications between RIM and Google with regard to such synchronization occurs over port 995 for POP3/SSL, ports 465 and 587 for SMTP/SSL, port 993 for IMAP/SSL and port 443 for Google's HTTP/SSL based custom subscription and notification protocol.

4. Google provides no customizations or specialized software or implementations beyond POP3/SSL, SMTP/SSL, IMAP/SSL, and Google's HTTP/SSL based custom subscription and notification protocol to support access to RIM email accounts in conjunction with services provided to users by RIM, nor is it using any software provided by RIM to support access to Google email accounts in conjunction with services provided to users by RIM.

5. Since October 1, 2005, approximately ___ U.S. Google e-mail accounts have been synchronized through RIM's BlackBerry Internet Service. The attached Ex. A lists the approximate number of U.S. Google e-mail accounts synchronized through RIM's BlackBerry Internet Service on a monthly / daily basis since October 1, 2005. As of the date that this agreement was signed, there are ___ Google e-mail accounts registered to perform synchronization with RIM BlackBerry enabled handhelds, of which approximately ___ are United States registered e-mail accounts. Since October 1, 2005, there have been ___ Google e-mail accounts registered to perform synchronization with RIM BlackBerry enabled handhelds, of which approximately ___ are United States registered e-mail accounts.

6. Google and the undersigned declare under penalty of perjury pursuant to 28 U.S.C. § 1746 that the foregoing is true and correct. This declaration was executed on April ____, 2008, in Mountain View, California.

Signature: _____

Name and Title: _____

Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043

EXHIBIT 'B'

LAW OFFICES
KEKER & VAN NEST
LLP
710 SANSOME STREET
SAN FRANCISCO, CA 94111-1704
TELEPHONE (415) 391-5400
FAX (415) 397-7188
WWW.KVN.COM

KHARI J. TILLERY
KTILLERY@KVN.COM

March 4, 2008

VIA FACSIMILE & U.S. MAIL

Geoffrey L. Smith, Esq.
McKool Smith
300 West 6th Street, Suite 1700
Austin, TX 78701

Re: *Subpoena in Visto Corporation v. Research In Motion Limited and Research In Motion Corporation, No. 2-06-CV-181 (E.D. Tex)*

Dear Mr. Smith:

It was a pleasure speaking with you on Friday. And thank you again for agreeing to extend our time to respond to the subpoena until March 10, 2008. I am writing now to confirm the details of our conversation.

You told us that the patents at issue in the *Visto v. RIM* litigation teach a system of synchronizing various data components of a user's account (email, calendar, etc.) between a server behind a firewall and handheld devices (e.g., smart phones). And it is your belief that RIM has software and/or hardware that synchronizes these data components between Google servers and RIM's BlackBerry handheld devices, and that Google may have information about this software and/or hardware.

During our conversation, we also discussed each of the seven document requests in the subpoena. The following is our understanding of what Visto is hoping to receive from Google with respect to each of the requests.

Request No. 1: Visto is not concerned about the specific location of Google's servers, but only whether some of the servers are located within the United States.

Request No. 2: Visto wants to know the total number of Google users that synchronize their data with RIM devices. As an example, you thought there might be a per customer synchronizing fee agreement between RIM and Google. As we explained, however, no such agreement exists.

Request No. 3: Visto wants design documents describing the software and/or hardware interface between RIM and Google, if any.

Request No. 4: Visto would like documents relating to software and/or hardware provided by RIM for instillation at Google, if any, but is not interested in RIM devices used by Google employees.

Requests Nos. 5-6: We could not understand from the subpoena what you meant by the term "protocol," as used in Requests Nos. 5-6, and you explained that this was a reference to email protocols such as Internet Message Access Protocol (IMAP) and Post Office Protocol (POP).

Request No. 7: You described this request as a "catch-all" request meant to capture any documents evidencing system configurations, if any, that allow Google servers to communicate with RIM's BlackBerry Internet Service (or BlackBerry Web Client), such as a firewall port.

Finally, you told us that you requested these same documents from RIM, but have not received a single responsive document. Nonetheless, you have neither moved to compel production of these documents from RIM nor noticed a 30(b)(6) deposition of RIM on these topics.

I will call you on March 5, 2008 to further discuss the subpoena and Google's response. In the meantime, if you have any questions or concerns, please do not hesitate to call me.

Very truly yours,

Khari J. Tillery

Khari J. Tillery

KJT/dbm

cc: Michael Page, Esq.
Ashok Ramani, Esq.

EXHIBIT 'C'

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VISTO CORPORATION,	§	
	§	**Case No.** 2-06CV-181
Plaintiff,	§	
	§	
v.	§	
	§	**Jury Trial Demanded**
RESEARCH IN MOTION LIMITED, and	§	
RESEARCH IN MOTION CORPORATION,	§	
	§	
Defendants.		

ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Visto Corporation ("Visto") alleges as follows for its Complaint against Defendants Research In Motion Limited and Research In Motion Corporation (collectively, "RIM"):

PARTIES

1. Visto is a corporation organized and existing under the laws of the State of Delaware with a place of business at 101 Redwood Shores Parkway, 4th Floor, Redwood City, California 94065

2. Research In Motion Limited is a corporation organized and existing under the laws of Ontario, Canada with a principal place of business at 295 Phillip Street, Waterloo, Ontario, Canada N2L 3W8, which is doing business and infringing Visto's patents in the Eastern District of Texas and elsewhere in the United States.

3 Research In Motion Corporation is a corporation organized and existing under the laws of the State of Delaware and registered to do business in the State of Texas with a

place of business at 122 West John Carpenter Parkway, Suite 430, Irving, Texas 75039, which is doing business and infringing Visto's patents in the Eastern District of Texas and elsewhere in the United States. Research In Motion Corporation is the United States distributor of Research In Motion Limited products and services.

JURISDICTION AND VENUE

4. This is a civil action for patent infringement arising under the patent laws of the United States, 35 U.S.C. § 271 *et seq*.

5. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. § 1331 and 28 U.S.C. § 1338(a).

6. RIM has sufficient contacts with the Eastern District of Texas to subject it to the personal jurisdiction of this Court for purposes of this Complaint, including, without limitation, acts of infringement of Visto's patents committed by RIM within this District.

7 Venue is proper in the Marshall Division of the Eastern District of Texas pursuant to 28 U.S.C. § 1391(b)-(c) and 28 U.S.C. § 1400(b).

GENERAL ALLEGATIONS

8. Established in 1996, Visto is a leading provider of personal and corporate wireless messaging solutions to mobile operators for personal and corporate use. Visto's inventions enable the deployment of a complete mobility solution, which provides secure access to and synchronization of the most widely used personal information management ("PIM") data over any network and on a broad array of mobile devices, such as personal digital assistants ("PDAs"), smartphones, and the like Visto has expended considerable resources in inventing and developing its inventions and protecting its rights therein.

9. Visto holds all right, title, and interest in and to United States Patent No.

6,085,192, entitled "System And Method For Securely Synchronizing Multiple Copies Of A Workspace Element In A Network" ("'192 patent"), which was duly and properly issued by the United States Patent and Trademark Office ("USPTO") on July 4, 2000 in the name of Daniel J. Mendez et al. A copy of the '192 patent is attached as Exhibit A to this Complaint. A Reexamination Certificate for the '192 patent was duly and properly issued by the USPTO on November 22, 2005. A copy of the Reexamination Certificate of the '192 patent is attached as Exhibit B to this Complaint.

10. Visto holds all right, title, and interest in and to United States Patent No. 6,151,606, entitled "System And Method For Using A Workspace Data Manager To Access, Manipulate And Synchronize Network Data" ("'606 patent"), which was duly and properly issued by the USPTO on November 21, 2000 in the name of Daniel J. Mendez. A copy of the '606 patent is attached as Exhibit C to this Complaint.

11 Visto holds all right, title, and interest in and to United States Patent No. 6,023,708, entitled "System And Method For Using A Global Translator To Synchronize Workspace Elements Across A Network" ("'708 patent"), which was duly and properly issued by the USPTO on February 8, 2000 in the name of Daniel J Mendez et al. A copy of the '708 patent is attached as Exhibit D to this Complaint.

12 Visto holds all right, title, and interest in and to United States Patent No 6,708,221, entitled "System And Method For Globally And Securely Accessing Unified Information In A Computer Network" ("'221 patent"), which was duly and properly issued by the USPTO on March 16, 2004 in the name of Daniel J Mendez et al A copy of the '221 patent is attached as Exhibit E to this Complaint. The '192 patent, '606 patent, '708 patent, and '221 patent are hereinafter referred to collectively as the "Patents-in-Suit".

13. RIM provides products and services that enable access to and/or synchronization of data in secure network environments (collectively, the "Accused Products"), including without limitation the products currently sold by RIM under the name BLACKBERRY®.

14. RIM infringes the Patents-in-Suit directly, contributorily and/or by active inducement by importing, manufacturing, using, marketing, distributing, selling, and/or supporting the Accused Products.

15. Upon information and belief, RIM's infringement of the Patents-in-Suit has been and continues to be willful.

COUNT I
Infringement of U.S. Patent No. 6,085,192

16. Visto incorporates paragraphs 1 through 15 as though fully restated herein.

17. RIM has infringed and continues to infringe the '192 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

18. RIM has contributorily infringed and/or induced others to infringe and continues to contributorily infringe and/or to induce others to infringe the '192 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

19. Upon information and belief, RIM's infringement of the '192 patent has been and continues to be willful.

20. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law. Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe

the '192 patent.

COUNT II
Infringement of U.S. Patent No. 6,151,606

21. Visto incorporates paragraphs 1 through 20 as though fully restated herein.

22. RIM has infringed and continues to infringe the '606 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

23. RIM has contributorily infringed and/or induced others to infringe and continues to contributorily infringe and/or to induce others to infringe the '606 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

24. Upon information and belief, RIM's infringement of the '606 patent has been and continues to be willful.

25. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe the '606 patent.

COUNT III
Infringement of U.S. Patent No. 6,023,708

26. Visto incorporates paragraphs 1 through 25 as though fully restated herein

27. RIM has infringed and continues to infringe the '708 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

28. RIM has contributorily infringed and/or induced others to infringe and

continues to contributorily infringe and/or to induce others to infringe the '708 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

29. Upon information and belief, RIM's infringement of the '708 patent has been and continues to be willful.

30. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law. Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe the '708 patent.

COUNT IV
Infringement of U.S. Patent No. 6,708,221

31. Visto incorporates paragraphs 1 through 30 as though fully restated herein.

32. RIM has infringed and continues to infringe the '221 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

33. RIM has contributorily infringed and/or induced others to infringe and continues to contributorily infringe and/or to induce others to infringe the '221 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

34. Upon information and belief, RIM's infringement of the '221 patent has been and continues to be willful

35. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law.

Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe the '221 patent.

PRAYER FOR RELIEF

WHEREFORE, Visto prays:

1. That RIM and its parents, affiliates, subsidiaries, officers, agents, servants, employees, attorneys, successors, and assigns, and all those persons in active concert or participation with them, or any of them, be preliminarily and permanently enjoined and restrained from making, importing, using, offering for sale, selling, or causing to be sold any product falling within, or designed to conduct a method falling within, the scope of any claim of the Patents-in-Suit, or otherwise infringing or contributing to or inducing infringement of any claim of the Patents-in-Suit;

2. That RIM and its parents, affiliates, subsidiaries, officers, agents, servants, employees, attorneys, successors, and assigns, and all those persons in active concert or participation with them, or any of them, be ordered to destroy or offer up to Visto for destruction any and all products within the scope of any claim of the Patents-in-Suit that are within RIM's possession, custody, or control;

3. That Visto be awarded its actual damages, including lost profits and price erosion, but in no case less than a reasonable royalty, to be assessed by or under the Court's discretion, adequate to compensate Visto for RIM's infringement of the Patents-in-Suit;

4. That Visto be awarded pre-judgment interest and post-judgment interest at the maximum rate allowed by law;

5 That the Court order an accounting for damages;

6. That the Court declare this to be an exceptional case pursuant to 35 U.S.C.

§ 285 and award Visto its attorneys' fees;

7. That the Court award enhanced damages pursuant to 35 U.S.C. § 284 due to the willful and wanton nature of RIM's infringement of the Patents-in-Suit;

8. That Visto be awarded costs of court; and

9 That Visto be awarded such other and further relief as the Court deems just and proper.

DEMAND FOR A JURY TRIAL

Pursuant to Rule 38 of the Federal Rules of Civil Procedure, Visto demands a trial by jury on all issues triable of right by a jury.

DATED: April 28, 2006

Respectfully submitted,

Sam F. Baxter, Lead Attorney
State Bar No. 01938000
McKOOL SMITH, P.C.
505 East Travis Street, Suite 105
Marshall, TX 75670
Telephone: (903) 927-2111
Facsimile: (903) 927-2622

Of Counsel:

Ronald S. Katz
Robert D. Becker
J. Bruce McCubbrey
Shawn G. Hansen
MANATT, PHELPS & PHILLIPS, LLP
1001 Page Mill Road, Building 2
Palo Alto, CA 94304
Telephone: (650) 812-1300
Facsimile: (650) 213-0260

Attorneys for Plaintiff
VISTO CORPORATION

EXHIBIT A

US006085192A

United States Patent [19]

Mendez et al.

[11] **Patent Number: 6,085,192**

[45] **Date of Patent: Jul. 4, 2000**

[54] **SYSTEM AND METHOD FOR SECURELY SYNCHRONIZING MULTIPLE COPIES OF A WORKSPACE ELEMENT IN A NETWORK**

[75] Inventors: **Daniel J. Mendez**, Mountain View; **Mark D. Riggins**, San Jose; **Prasad Wagle**, Santa Clara; **Christine C. Ying**, Foster City, all of Calif.

[73] Assignee: **RoamPage, Inc.**, Mountain View, Calif.

[21] Appl. No.: **08/835,997**

[22] Filed: **Apr. 11, 1997**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/10**; 707/203; 707/104; 707/1; 707/9; 707/10

[58] **Field of Search** 707/203, 104, 707/1, 9, 10

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,831,582	5/1989	Miller et al.	707/104
4,875,159	10/1989	Cary et al.	364/200
5,263,157	11/1993	Janis	707/9
5,386,564	1/1995	Shearer et al.	395/650
5,581,749	12/1996	Hossain et al.	707/1
5,600,834	2/1997	Howard	395/617
5,613,012	3/1997	Hoffman et al.	382/115
5,627,658	5/1997	Connors et al.	358/407
5,666,553	9/1997	Crozier	395/803
5,682,524	10/1997	Freund et al.	395/605
5,684,990	11/1997	Boothby	707/203
5,701,423	12/1997	Crozier	395/335
5,706,502	1/1998	Foley et al.	707/10
5,710,918	1/1998	Lagarde et al.	395/610
5,715,403	2/1998	Stefik	705/44
5,721,908	2/1998	Lagarde et al.	395/610
5,721,914	2/1998	DeVries	395/615
5,745,360	4/1998	Leone et al.	364/140
5,757,916	5/1998	MacDoran et al.	380/25
5,778,346	7/1998	Frid-Nielsen et al.	395/208
5,790,425	8/1998	Wagle	364/551.01
5,790,790	8/1998	Smith et al.	395/200.36
5,799,318	8/1998	Cardinal et al.	707/104

OTHER PUBLICATIONS

Network Firewalls, Bellovin et al., IEEE Communications Magazine, pp. 50–57, Sep. 1994.

IpAccess—An Internet Service Access System for Firewall Installations, Steffen Stempel, 1995 IEEE, pp. 31–41, Feb. 1995.

Web traffic characterization: an assessment of the impact of caching documents from NCSA's web server, Braun et al., Computer Networks and ISDN Systems, V. 28, pp. 37–51, Jul. 1995.

Security for Infinite Networks, Nelson et al., Aug. 1995 IEEE, pp. 11–19.

DEsigning an Academic Firewall: Policy, Practice, and Experience with SURF, Greenwald et al., Proceedings of SNDSS '96, Feb. 1996 IEEE, pp. 79–92.

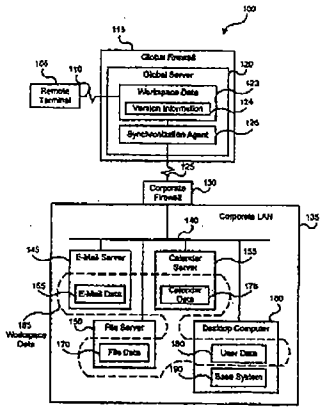
(List continued on next page.)

Primary Examiner—Wayne Amsbury
Assistant Examiner—Shahid Alam
Attorney, Agent, or Firm—Graham & James LLP

[57] **ABSTRACT**

A system includes a general synchronization module at the client site for operating within a first firewall and for examining first version information to determine whether a first workspace element has been modified. The system further includes a synchronization agent at a global server for operating outside the first firewall and for forwarding to the general synchronization module second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified. A synchronization-start module is maintained at the client site for operating within the first firewall and for securely initiating the general synchronization module and the synchronization agent when predetermined criteria have been satisfied. The system further includes means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information, and means for storing the preferred version at the first store and at the second store

25 Claims, 6 Drawing Sheets



OTHER PUBLICATIONS

C–HTTP—The Development of a Secure, Closed HTTP–based Network on the Internet, Kiuchi et al., Proceedings of SNDSS '96, Feb. 1996 IEEE, pp. 64–75.

Establishing Firewall Policy, S Cobb, Director of Special Projects National Computer Security Associations, Florida Office, ISBN#0–7803–3268–7, Jun. 1996, pp. 198–205.

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome– Jul. 1997, URL=http://www.bns.it/emware/BookmarkTranslator–uk.htm, pp. 1–4

Article by Bellovin et al , entitled: "Network Firewalls"Published by IEEE Communications Magizine Sep. 1994, pp.50–57

Article by Steffen Stempel, entitled: "IPAccess–An Internet Service Access System for Firewall Installations"Published by IEEE Communications Magazine a Feb 16, 1995, pp.31–41.

Article by Braun et al , entitled:"Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server"Published by Elservier Science B.V. 1995 pp. 37–51.

Article by Nelson et al., entitled: "Security for Infinite Networks"Published by IEEE Communications Magazine on Aug. 22, 1995, pp.11–19.

Article by Greenwald et al , "Designing an Academic Firewall: Policy, Pratice, and Experience with SURF"Published by IEEE Communications Magazine on Feb. 22, 1996, pp. 79–92.

Article by Kiuchi et al , entitled: "C–HTTP–The Development of a Secure, Closed HTTP–based Network on the Internet"Published by IEEE Proceedings of SNDSS on Feb. 22, 1996, pp 64–75

Article by S Cobb, entitled: "Establishing Firewall Policy-"Published by National Computer Security Assn. on Jun 25–27, 1996, pp. 198–205.

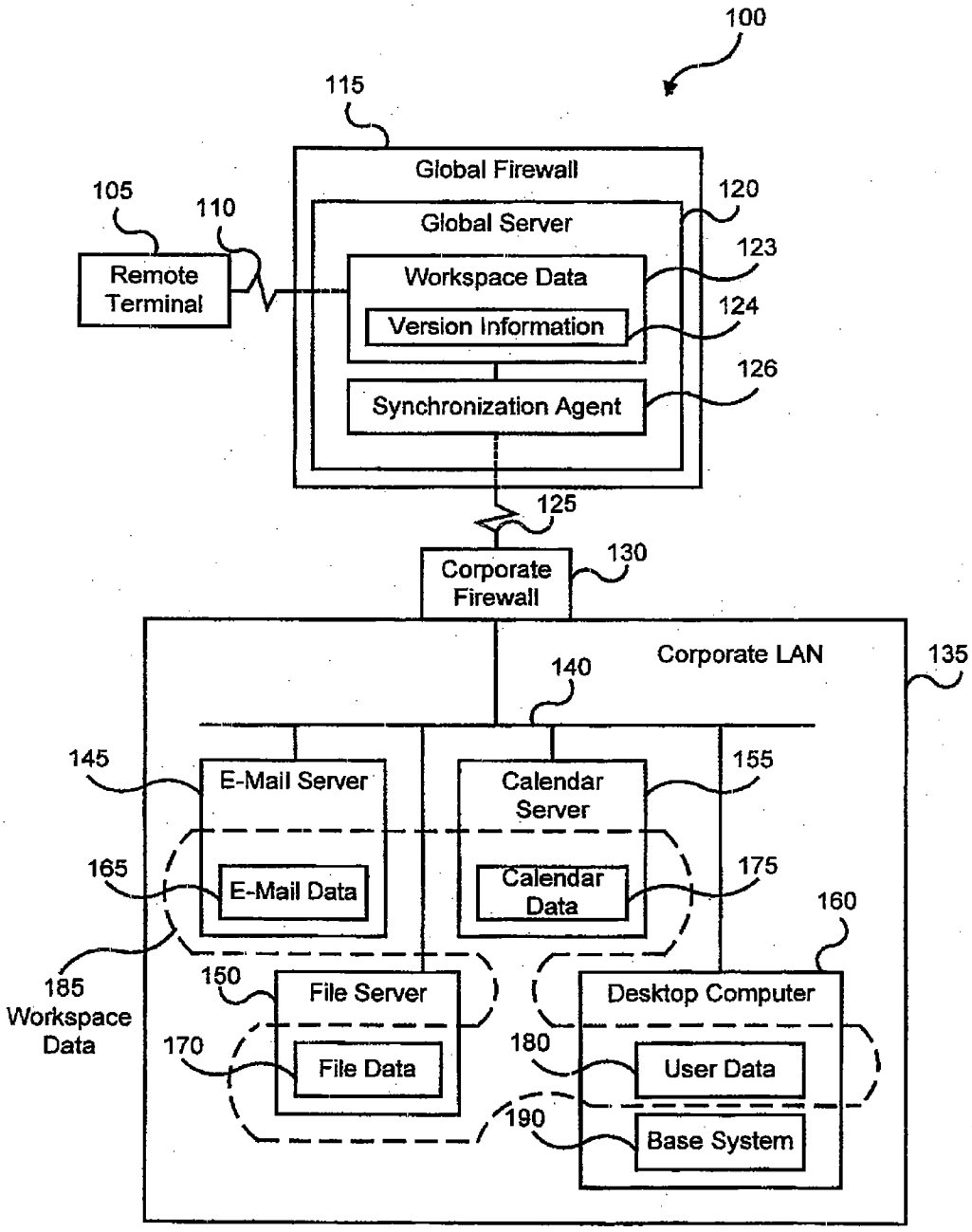

100
115
Global Firewall
105
110
120
Global Server
123
Remote Terminal
Workspace Data
124
Version Information
126
Synchronization Agent
125
130
Corporate Firewall
Corporate LAN
135
140
145
E-Mail Server
155
Calendar Server
165
E-Mail Data
175
Calendar Data
160
185 Workspace Data
150
File Server
Desktop Computer
170
File Data
180
User Data
190
Base System

FIG. 1

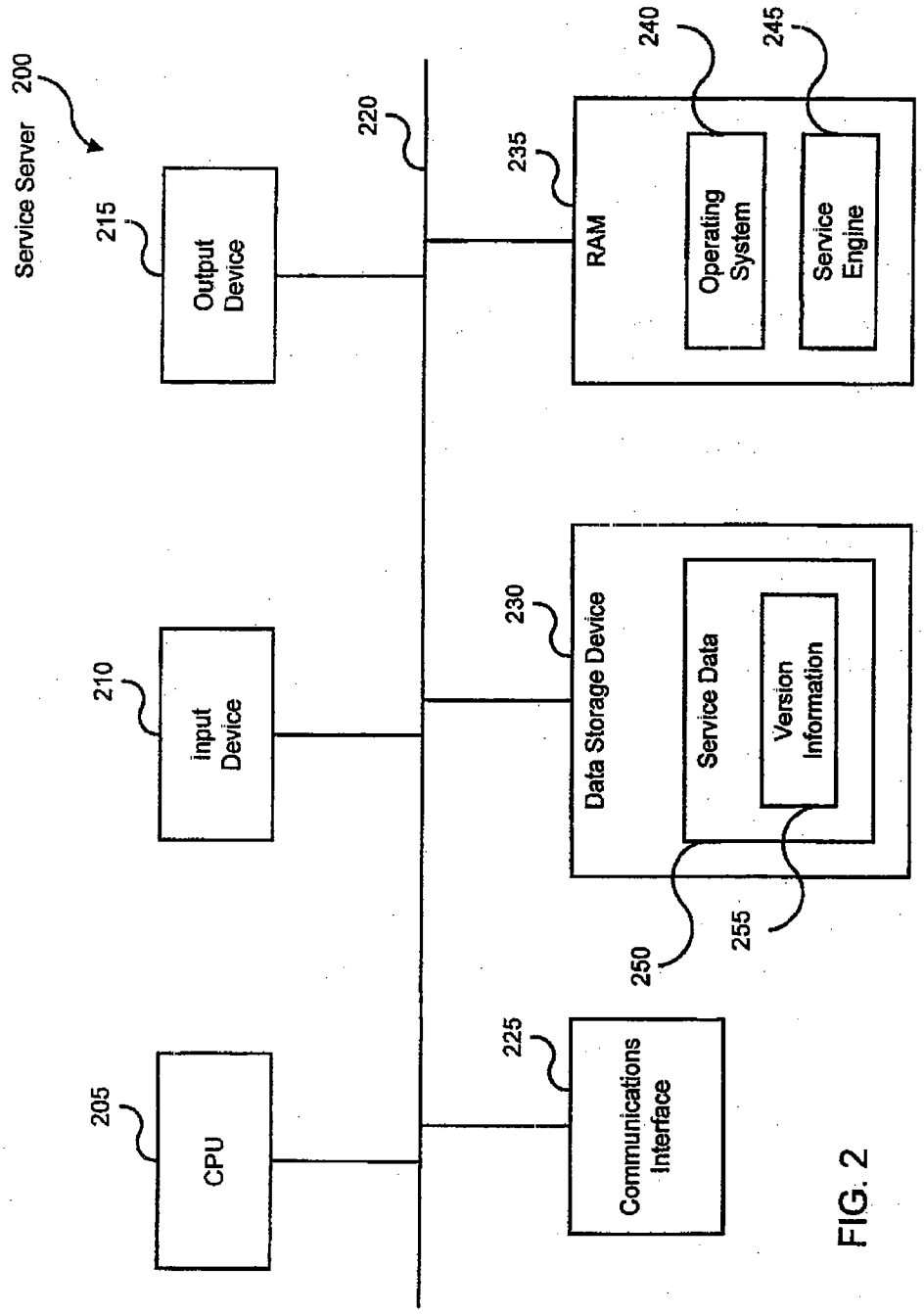

Service Server
200
220
215
Output Device
235
RAM
240
Operating System
245
Service Engine
210
Input Device
230
Data Storage Device
Service Data
Version Information
250
255
205
CPU
225
Communications Interface

FIG. 2

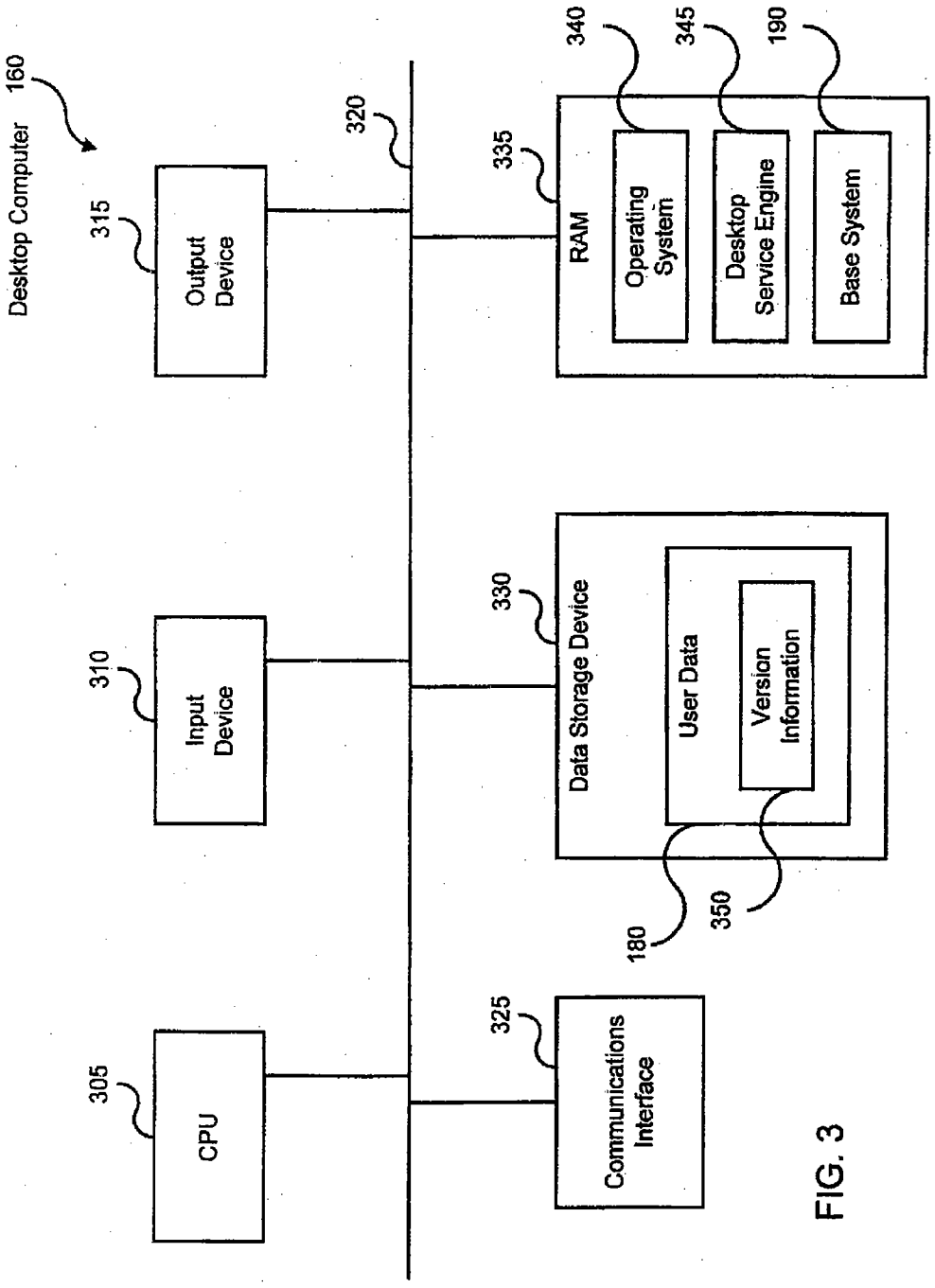

Desktop Computer
160
315
Output Device
320
335
RAM
340
Operating System
345
Desktop Service Engine
190
Base System
310
Input Device
330
Data Storage Device
User Data
Version Information
180
350
305
CPU
325
Communications Interface

FIG. 3

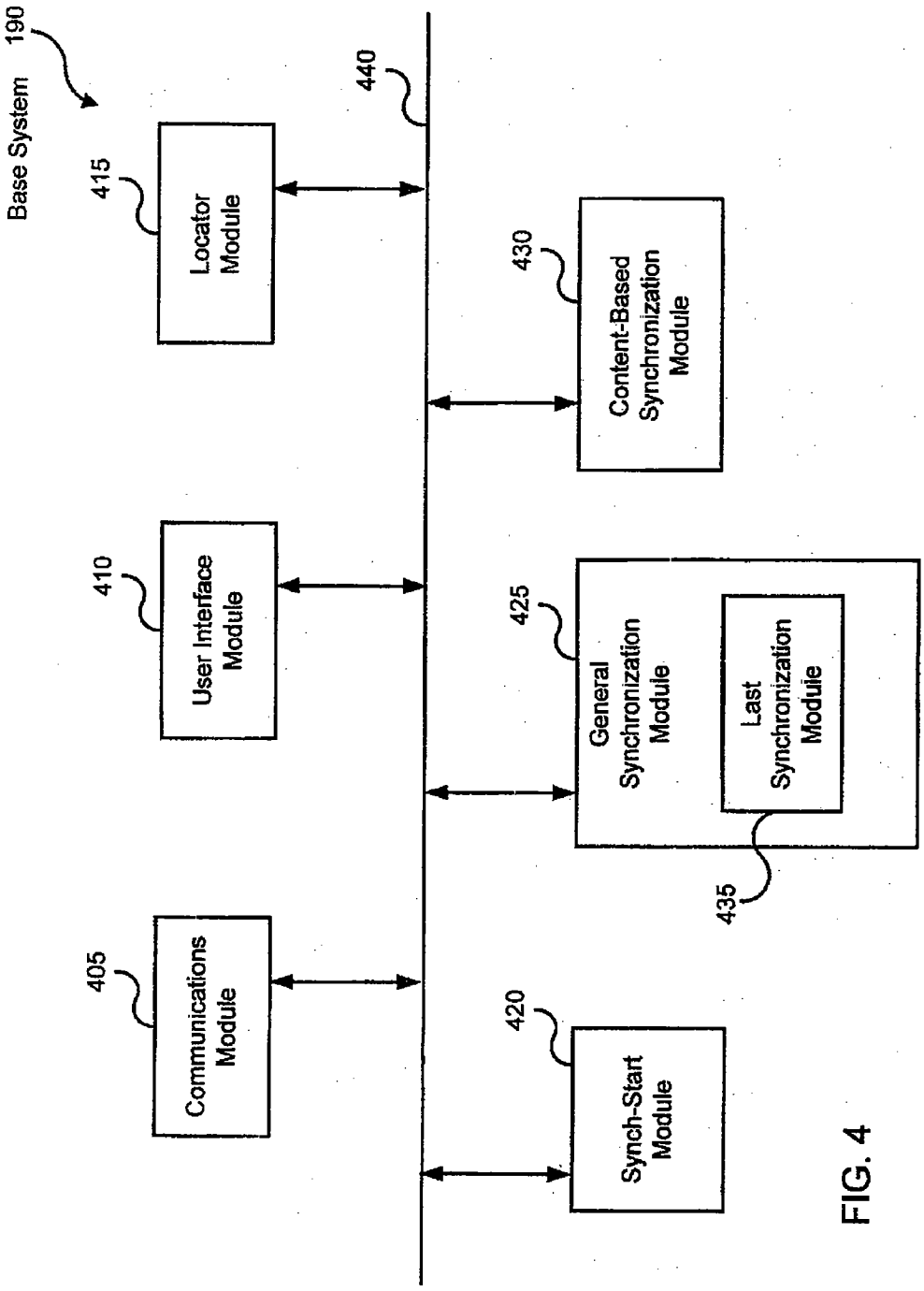

Base System
190
440
415
Locator Module
430
Content-Based Synchronization Module
410
User Interface Module
425
General Synchronization Module
Last Synchronization Module
435
405
Communications Module
420
Synch-Start Module

FIG. 4

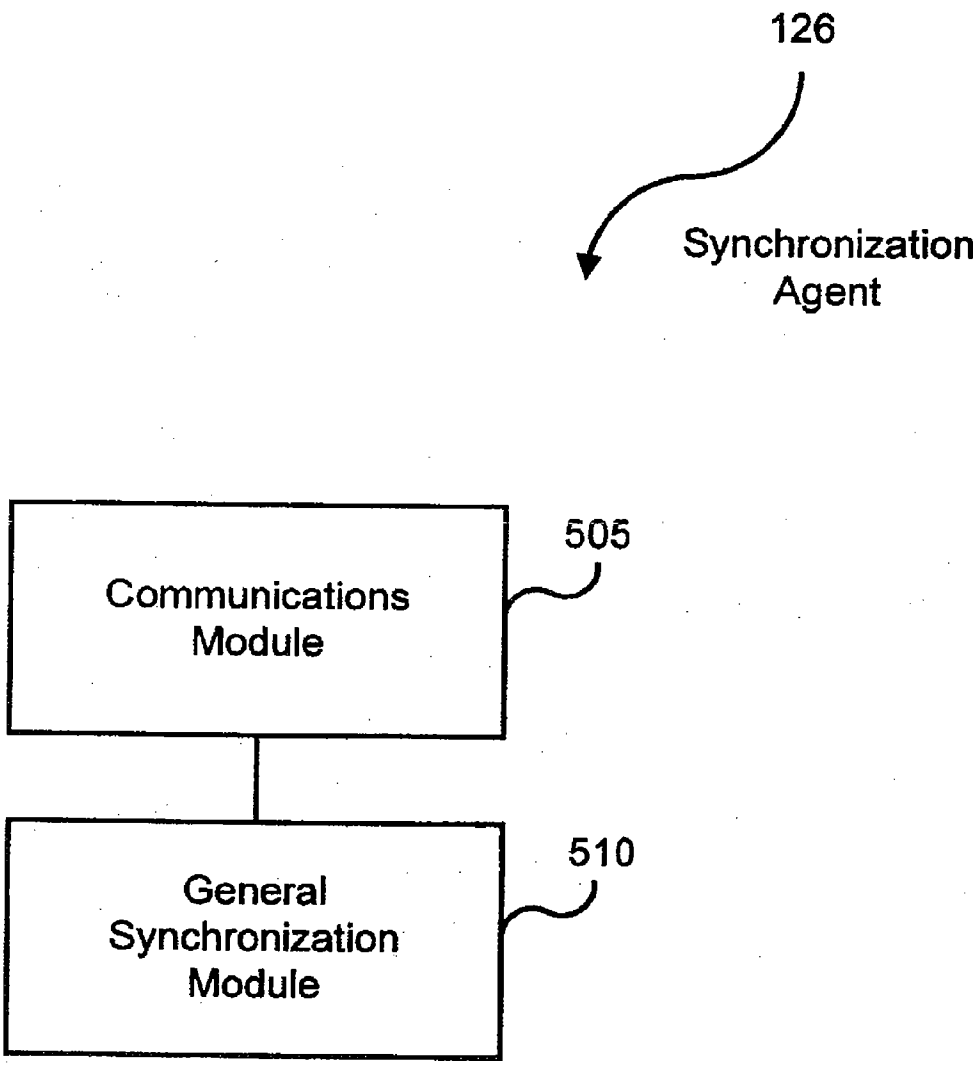

126
Synchronization Agent
Communications Module
505
General Synchronization Module
510

FIG. 5

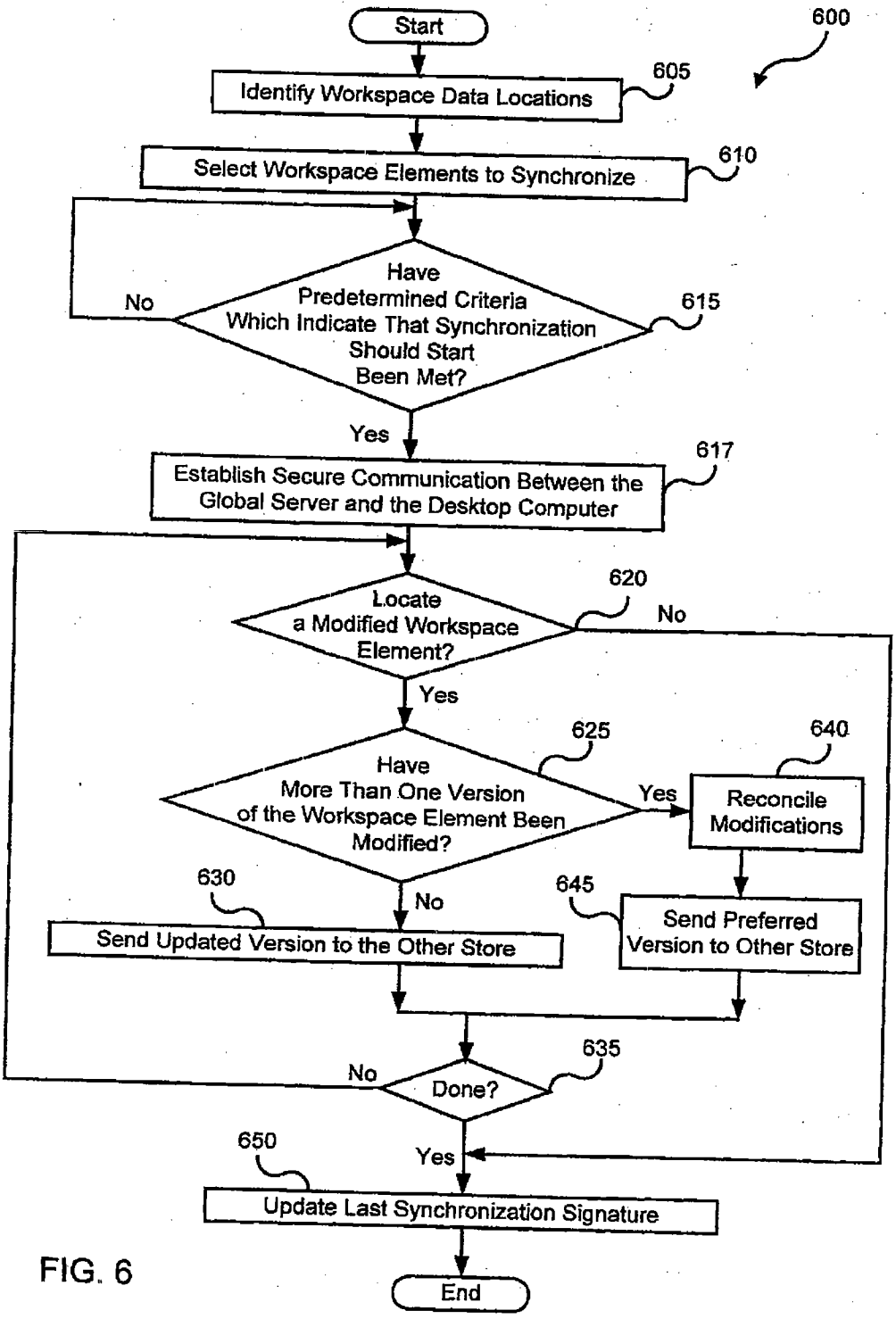

600
Start
Identify Workspace Data Locations
605
Select Workspace Elements to Synchronize
610
Have Predetermined Criteria Which Indicate That Synchronization Should Start Been Met?
615
No
Yes
Establish Secure Communication Between the Global Server and the Desktop Computer
617
Locate a Modified Workspace Element?
620
No
Yes
Have More Than One Version of the Workspace Element Been Modified?
625
Yes
Reconcile Modifications
640
No
630
Send Updated Version to the Other Store
645
Send Preferred Version to Other Store
Done?
635
No
Yes
650
Update Last Synchronization Signature
End

FIG. 6

SYSTEM AND METHOD FOR SECURELY SYNCHRONIZING MULTIPLE COPIES OF A WORKSPACE ELEMENT IN A NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to co-pending patent application entitled "System and Method for Globally Accessing Computer Services," Ser. No 08/766,307, now pending, filed on Dec. 13, 1996, by inventors Mark D. Riggins, R. Stanley Bailes, Hong Q. Bui, David J. Cowan, Daniel J. Mendez, Mason Ng, Sean Michael Quinlan, Prasad Wagle, Christine C. Ying, Christopher R. Zuleeg and Joanna A. Aptekar-Strober; and to co-pending patent application entitled "System and Method for Enabling Secure Access to Services in a Computer Network," Ser. No. 08/841,950, now pending, filed on Apr. 8, 1997, by inventor Mark Riggins, both of which are hereby incorporated by reference These related applications have been commonly assigned to RoamPage, Inc

BACKGROUND OF THE INVENTION

1 Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for securely synchronizing multiple copies of a workspace element such as a file in a secure network.

2. Description of the Background Art

Data consistency is a significant concern for computer users. For example, when maintaining multiple independently-modifiable copies of a document, a user risks using an outdated version. Further, by the time the user notices the inconsistency, interparty miscommunication or data loss may have resulted. The user must then spend more time attempting to reconcile the inconsistent versions

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft

Therefore, a system and method are needed for providing users with data consistency, and more particularly for synchronizing multiple copies of a workspace element such as a document in the secure network environment.

SUMMARY OF THE INVENTION

The present invention provides a system and method for synchronizing multiple copies of a workspace element in a secure network environment. The secure network environment includes a global server connected to multiple clients. Using the present system and method, the clients automatically synchronize workspace data between multiple sites, independent of whether the sites are protected by site firewalls

The present system includes a general synchronization module at the client site for operating within a first firewall and for examining first version information to determine whether a first workspace element has been modified. The system further includes a synchronization agent at the global server for operating outside the first firewall and for forwarding to the general synchronization module second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified. A synchronization-start module. at the client site operates within the first firewall and initiates the general synchronization module and the synchronization agent when predetermined criteria have been satisfied. The system further includes means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information, and means for storing the preferred version at the first store and at the second store.

The system further handles the case when both the workspace element and the copy have been modified independently since the last date and time of synchronization. Accordingly, a content-based synchronization module performs a responsive action such as determined a preferred version or storing both the first workspace element and the copy at both the first store and at the second store

The present method includes the steps of generating first examination results by examining first version information, which indicates whether a first workspace element stored at a first store within a firewall has been modified; and generating second examination results by examining second version information which indicates whether an independently-modifiable copy of the first workspace element, the copy being stored at a second store outside the firewall, has been modified. The present method further includes the steps of initiating synchronization from within the firewall when predetermined criteria have been satisfied; generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and storing the preferred version at the first store and at the second store.

The system and method advantageously use a trusted third party to enable the synchronization of workspace data among multiple sites. Accordingly, a client user who maintains a work site, a home site, an off-site and the global server site can synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which controls when the synchronizationstart module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, synchronization can occur over any distance. Since synchronization is initiated from within the firewall, the typical firewall, which prevents in-bound communications, does not act as an impediment to workspace data synchronization Also, since the user's preferences may be previously set, the present system and method may operate unattended by the client user

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a secure data-synchronizing network in accordance with the present invention;

FIG 2 is a block diagram illustrating details of a FIG. 1 service server;

FIG. 3 is a block diagram illustrating details of the FIG. 1 desktop computer;

FIG. 4 is a block diagram illustrating details of the FIG 3 base system;

FIG. 5 is a block diagram illustrating details of the FIG. 1 synchronization agent; and

FIG. 6 is a flowchart illustrating a method for synchronizing multiple copies of a workspace element in a secure network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG 1 is a block diagram illustrating a secure data-synchronizing network **100**, comprising a first site such as a remote computer terminal **105** coupled via a communications channel **110** such as the Internet to a global server **120**. The global server **120** is in turn coupled via a communications channel **125** such as the Internet to a second site such as a corporate Local Area Network (LAN) **135**. The global server **120** is protected by a global firewall **115**, and the corporate LAN **135** is protected by a corporate firewall **130**.

The corporate LAN **135** includes a corporate signal bus **140** coupling the corporate firewall **130** to an e-mail server **145** having e-mail data **165**, to a file server **150** having file data **170**, to a calendar server **155** having calendar data **175** and to a desktop computer **160** having user data **180**. It will be appreciated that the e-mail data **165**, file data **170**, calendar data **175** and user data **180** or portions thereof may be stored at different locations such as locally on the desktop computer **160**. It will be further appreciated that the e-mail data **165**, file data **170**, calendar data **175** and user data **180** are exemplary and collectively referred to herein as "workspace data" **185**. Those skilled in the art will recognize that "workspace data" may include other types of data such as application programs. It will be further appreciated that the e-mail data **165**, file data **170**, calendar data **175** and user data **180** may each be divided into workspace elements, wherein each workspace element is identified by particular version information **255** (described below with reference to FIG 2). Accordingly, each e-mail, file, calendar, etc may be referred to as "a workspace element in workspace data."

An independently modifiable copy of the workspace data **185**, referred to herein as workspace data **123**, is stored on the global server **120** for easy access by a user from the remote terminal **105**. Being a copy, the workspace data **123** includes independently modifiable copies of each workspace element in workspace data **185** and an independently modifiable copy of version information **255** (FIG. 2), referred to herein as version information **124**.

Network **100** further comprises synchronization means, which includes a base system **190** stored within the corporate LAN **135** and for example on the desktop computer **160** and further includes a synchronization agent **126** stored outside the corporate firewall **130** and preferably on the global server **120**. The base system **190** and the synchronization agent **126** cooperate to synchronize the workspace data **185** with the workspace data **123**. Generally, the base system **190** manages the workspace data **185** within the corporate LAN **135** and the synchronization agent **126** manages the workspace data **123** within the global server **120**. As described in greater detail below with reference to FIG. **4**, the base system **190** preferably initiates and controls data synchronization. Other components and functions of the global server **120** are described in the cross-referenced patent applications which are herein incorporated by reference.

The remote terminal **105** may include a smart telephone or a Personal Data Assistant (PDA) such as the PalmPilot system by the U.S. Robotics, Inc. Although not shown, the remote terminal **105** may include a second base system similar to the base system **190**, which is described with greater detail with reference to FIG. **4**. Accordingly, the second base system on the remote terminal **105** would cooperate with the synchronization agent **126** to synchronize the workspace data stored on the remote terminal **105** with the workspace data **123** stored on the global server **120**. As

with the corporate LAN, the second base system on the remote terminal **105** would preferably initiate and control data synchronization with the global server **120** for the same reasons discussed below. Workspace data on the remote terminal **105** would thus be synchronized with the workspace data **123** and with the workspace data **185**.

FIG. **2** is a block diagram illustrating details of a service server **200**, wherein each of the e-mail server **145**, the file server **150**, the calendar server **155** and the desktop computer **160** is an instance thereof. Service server **200** includes a Central Processing Unit (CPU) **205** such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor. An input device **210** such as a keyboard and mouse and an output device **215** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **220** to CPU **205**. A communications interface **225** (such as an Ethernet port), a data storage device **230** (such as read only memory or a magnetic disk), and Random-Access Memory (RAM) **235** are further coupled via signal bus **220** to the CPU **205**.

An operating system **240** includes a program for controlling processing by the CPU **205**, and is typically stored in the data storage device **230** and loaded into the RAM **235** for execution. A service engine **245** includes a program for performing a particular service such as maintaining an e-mail data base, a calendar data base, a bookmarks data base or another file data base, and may be also stored in the data storage device **230** and loaded into the RAM **235** for execution. To perform a service, the service engine **245** operates on service data **250** (e.g., the e-mail data **165**, the file data **170**, the calendar data **175** or the user data **180**), which is typically stored in the data storage device **250**. The service data **250** includes version information **255** indicating the date and time of the last modification. The service engine **245** operates to update the version information **255** whenever modifications are made. It will be appreciated that the portion of memory in the data storage device **250** which contains the service data **250** is referred to as the service "store."

FIG. **3** is a block diagram illustrating details of the desktop computer **160**, which includes a CPU **305**, an input device **310**, an output device **315**, a communications interface **325**, a data storage device **330** and RAM **335**, each coupled to a signal bus **320**.

An operating system **340** includes a program for controlling processing by the CPU **305**, and is typically stored in the data storage device **330** and loaded into the RAM **335** for execution. A desktop service engine **345** (i.e., a particular service engine **245**, FIG. 2) includes a service program for managing user data **180** (i.e., particular service data **250**, FIG. 2) which includes version information **350** (i.e., particular version information **255**, FIG. 2). The desktop service engine **345** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution. The user data **180** may be stored in the data storage device **330**. As stated above with reference to FIG. **1**, the base system **190** operates to synchronize the workspace data **185** (which includes user data **180**) with the workspace data **123**. The base system **190** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution.

FIG. **4** is a block diagram illustrating details of the base system **190**, which includes a communications module **405**, a user interface module **410**, a locator module **415**, a synchronization-start ("synch-start") module **420**, a general synchronization module **425** and a content-based synchronization module **430**. For simplicity, each module is illustrated as communicating with one another via a signal bus **440**.

The communications module **405** includes routines for compressing data, and routines for communicating via the communications interface **325** (FIG. **3**) with the synchronization agent **126** (FIG. **1**). The communications module **405** may further include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the corporate firewall **130** and through the global firewall **126**. Examples of communications modules **405** may include TCP/IP stacks or the AppleTalk® protocol

The user interface **410** includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface **410** operates in coordination with the other desktop computer **160** components as described herein.

The locator module **415** includes routines for identifying the memory locations of the workspace elements in the workspace data **185** and the memory locations of the workspace elements in the workspace data **123**. Workspace element memory location identification may be implemented using intelligent software, i.e, preset memory addresses or the system's registry, or using dialogue boxes to query a user. Accordingly, the locator module **415** determines the memory addresses of the workspace elements in the e-mail data **165**, the workspace elements in the file data **170**, the workspace elements in the calendar data **175** and the workspace elements in the user data **180** as well as the memory addresses of the corresponding workspace elements in the workspace data **123** It will be appreciated that the locator module **415** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **120** to maintain updated memory locations of workspace elements.

The synchronization-start module **420** includes routines for determining when to initiate synchronization of workspace data **123** and workspace data **185**. For example, the synchronization-start module **420** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria The synchronization-start module **420** initiates data synchronization by instructing the general synchronization module **425** to begin execution of its routines. It will be appreciated that communications with synchronization agent **126** preferably initiate from within the corporate LAN **1135**, because the typical corporate firewall **130** prevents in-bound communications and allows outbound communications.

The general synchronization module **425** includes routines for requesting version information **124** from the synchronization agent **126** (FIG. **1**) and routines for comparing the version information **255** against a last synchronization signature **435** such as a last synchronization date and time to determine which versions have been modified. The general synchronization module **425** further includes routines for comparing the version information **124** and the version information **255** to determine if only one or both versions of a particular workspace element have been modified and routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include forwarding the modified version (as the preferred version) of a workspace element in workspace data **185** or forwarding just a compilation of the changes to the other store(s). Other appropriate synchronizing responsive actions may include, if reconciliation between two modified versions is needed, then instructing the content-based synchronization module **430** to execute its routines which are described below.

It will be appreciated that the synchronization agent **126** preferably examines the version information **124** and forwards only the version information **124** determined to be modified since the last synchronization signature **435**. This technique makes efficient use of processor power and avoids transferring unnecessary data across the communications channel **125**. The general synchronization module **425** in the corporate LAN **135** accordingly compares the received version information **124** with the version information **255** to determine if reconciliation is needed Upon completion of the data synchronization, the general synchronization module **425** updates the last synchronization signature **435**.

The content-based synchronization module **430** includes routines for reconciling two or more modified versions in workspace data **123, 185** of the same workspace element. For example, if the original and the copy of a user workspace element have both been modified independently since the last synchronization, the contentbased synchronization module **430** determines the appropriate responsive action. The content-based synchronization module **430** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e, by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores

FIG. **5** is a block diagram illustrating details of the synchronization agent **126**, which includes a communications module **505** (similar to the communications module **405** described above with reference to FIG. **4**) and a general synchronization module **510** (similar to the general synchronization module **425** described above also with reference to FIG. **4**). The communications module **505** includes routines for compressing data, and routines for communicating via the communications channel **125** with the base system **190**. The communications module **505** may further include routines for establishing a secure communications channel through the global firewall **126** and through the corporate firewall **130**.

The general synchronization module **510** includes routines for comparing the version information **124** with the last synchronization signature **435**, and routines for forwarding to the general synchronization module **425** version information **124** determined to be modified. The general synchronization module **510** may either maintain its own last synchronization signature **435** copy (not shown). Alternatively, the request to synchronize from the base system **190** may include a copy of the last synchronization signature **435** The general synchronization module **510** further includes routines for receiving preferred versions of workspace data **185** workspace elements from the general synchronization module **425**, and routines for forwarding preferred versions of workspace data **123** workspace elements to the general synchronization module **425**.

FIG. **6** is a flowchart illustrating a method **600** for synchronizing multiple copies of workspace data **123, 185** in a secure network **100** Method **600** begins with locator module **415** in step **605** identifying the memory locations of the workspace elements in workspace data **123, 185**. As stated above, workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The user interface module **410** in step **610** enables selection of the workspace elements in workspace data **123, 185** to be synchronized by the general synchronization module **425**.

The synchronization-start module **420** in step **615** determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step **610** should start. If not, then method **600** loops back to step **615**. Otherwise, the communications module **405** and communications module **505** in step **617** establish a secure communications channel between the global server **120** and the desktop computer **160** The general synchronization module **510** in step **620** compares the version information **124** of each of the selected workspace elements in workspace data **123** against the last synchronization signature **435** to determine modified workspace elements, and forwards the version information **124** of workspace elements determined to be modified to the general synchronization module **425**. Further, the general synchronization module **425** in step **620** compares the version information **255** of each selected workspace element in the workspace data **185** against the last synchronization signature **435** to locate modified workspace elements. In this embodiment, a workspace element has been modified if the date and time of last modification is after the date and time of last synchronization.

If no modified workspace elements in workspace data **123** or in workspace data **185** are located, then the general synchronization modules **425** and **510** in step **650** update the last synchronization signature **435** and method **600** ends. Otherwise, the general synchronization module **425** in step **625** determines whether more than one version of the same workspace element has been modified since the last synchronization

If only one version has been modified, then the corresponding general synchronization module **425** or **510** in step **630** forwards the updated preferred version of the workspace element to the other store, and then in step **635** determines whether all workspace elements selected in step **610** have been examined If so, then method **600** jumps to step **650**. Otherwise, then method **600** returns to step **620**.

If more than one version has been modified, then the general synchronization module **425** in step **640** instructs the content-based synchronization module **430** to reconcile the modified versions. Reconciliation may include requesting instructions from the user or, based on preselected preferences, performing responsive actions such as storing both versions at both stores.

General synchronization module **425**, **510** in step **645** sends the preferred version of the workspace element or just a compilation of the changes to the other store. That is, if the preferred version is a workspace element in the workspace data **185**, then general synchronization module **425** sends the preferred version or the changes to general synchronization module **510** to update the outdated workspace element in the workspace data **123**. If the preferred version is a workspace element in the workspace data **123**, then the general synchronization module **510** sends the preferred version or the changes to the general synchronization module **425** to update the outdated workspace element in the workspace data **185**. Method **600** then jumps to step **635**.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, although the global server **120** is illustrated as a single device, the global server **120** may include several computers networked together. Although not described in great detail, the remote terminal **105** can synchronize copies of workspace elements stored on it with workspace elements of workspace data **123** stored on the global server **120**. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A computer-based method comprising the steps of:

(a) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

(b) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

(c) initiating steps (a) and (b) from within the firewall when predetermined criteria have been satisfied;

(d) generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and

(e) storing the preferred version at the first store and at the second store

2. The method of claim **1** wherein the second store is on a global server outside the firewall and which is protected by a global firewall.

3. The method of claim **1** wherein the first version information includes the date and time the first workspace element was last modified and the second version information includes the date and time the copy was last modified.

4. The method of claim **3** wherein generating the first examination results includes the step of comparing the first version information against a date and time of last synchronization.

5 The method of claim **3** wherein generating the second examination results includes the step of comparing the second version information against a date and time of last synchronization

6. The method of claim **1** further comprising, before generating the first examination results, the step of updating the first version information whenever the first workspace element is modified

7. The method of claim **1** further comprising, before generating the second examination results, the step of updating the second version information whenever the copy is modified.

8. The method of claim **1** wherein if only one of the first workspace element and the copy has been modified, then the step of generating includes selecting the one as the preferred version.

9. The method of claim **1** further comprising the step of locating the first workspace element, the first version information, the copy and the second version information

10. A system comprising:

a general synchronization module for operating within a first firewall and for examining first version information to determine whether a first workspace element has been modified;

a synchronization agent for operating outside the first firewall and for forwarding to the general synchronization module second version information which indi-

cates whether an independently modifiable copy of the first workspace element has been modified;

a synchronization-start module for operating within the first firewall and for initiating the general synchronization module and the synchronization agent when predetermined criteria have been satisfied;

means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information; and

means for storing the preferred version at the first store and at the second store.

11. The system of claim **10** further comprising a communications module for communicating through the first firewall.

12. The system of claim **10** wherein the synchronization agent and the second store are on a global server which is protected by a global firewall.

13. The system of claim **12** further comprising a communications module for communicating through the first firewall and through the global firewall.

14. The system of claim **10** wherein the first version information includes the date and time the first workspace element was last modified and the second version information includes the date and time the copy was last modified.

15. The system of claim **14** wherein the general synchronization module compares the first version information against a date and time of last synchronization.

16. The system of claim **14** wherein the synchronization agent compares the second version information against the date and time of last synchronization.

17. The system of claim **10** further comprising means for updating the first version information whenever the first workspace element is modified.

18. The system of claim **10** further comprising means for updating the second version information whenever the copy is modified.

19. The system of claim **10** wherein if only one of the first workspace element and the copy has been modified, then the means for generating selects the one as the preferred version

20. The system of claim **10** further comprising a locator module for locating the first store, the first workspace element, the first version information, the second store, the copy and the second version information

21 A system comprising:

first means for generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

second means for generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

means for initiating the first and second means from within the firewall when predetermined criteria have been satisfied;

means for generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and

means for storing the preferred version at the first store and at the second store.

22. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

(a) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

(b) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

(c) initiating steps (a) and (b) from within the firewall when predetermined criteria have been satisfied;

(d) generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and

(e) storing the preferred version at the first store and at the second store.

23 A computer-based method comprising the steps of:

(a) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

(b) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

(c) initiating steps (a) and (b) from within the firewall when predetermined criteria have been satisfied;

(d) determining based on the first and second examination results that both the first workspace element and the copy have been modified; and

(e) storing both the first workspace element and the copy at the first store and at the second store.

24. A system comprising:

first means for generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

second means for generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

means for initiating the first and second means from within the firewall when predetermined criteria have been satisfied;

means for determining based on the first and second examination results that both the first workspace element and the copy have been modified; and

means for storing both the first file and the copy at the first store and at the second store.

25. A system comprising:

a global server for operating outside a firewall and including memory for storing first workspace data and corresponding first version information; and

a synchronization agent for managing the first workspace data and the corresponding first version information and for communicating with remote clients; and

a remote client for operating within the firewall and including memory for storing second workspace data and corresponding second version information;

means for cooperating with the synchronization agent to synchronize the first workspace data with the second workspace data by examining the first version information and the second version information; and

a synchronization-start module for initiating workspace data synchronization between the global server and the remote client

* * * * *

EXHIBIT B

US006085192C1

(12) EX PARTE REEXAMINATION CERTIFICATE (5234th)
United States Patent
Mendez et al.

(10) Number: US 6,085,192 C1
(45) Certificate Issued: Nov. 22, 2005

(54) **SYSTEM AND METHOD FOR SECURELY SYNCHRONIZING MULTIPLE COPIES OF A WORKSPACE ELEMENT IN A NETWORK**

(75) Inventors: **Daniel J. Mendez**, Mountain View, CA (US); **Mark D. Riggins**, San Jose, CA (US); **Prasad Wagle**, Santa Clara, CA (US); **Christine C. Ying**, Foster City, CA (US)

(73) Assignee: **Visto Corporation**, Redwood Shores, CA (US)

Reexamination Request:
No. 90/007,093, Jun 18, 2004

Reexamination Certificate for:
Patent No.: **6,085,192**
Issued: **Jul. 4, 2000**
Appl. No.: **08/835,997**
Filed: **Apr. 11, 1997**

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/10**; 707/1; 707/9; 707/10; 707/104; 707/203
(58) **Field of Search** ..... 707/1–4, 9–10, 707/203, 200, 101, 102, 103 R; 715/511; 709/201–202, 248–249

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,652,698	A	3/1987	Hale et al.
4,714,995	A	12/1987	Materna et al.
4,882,752	A	11/1989	Lindman et al
4,897,781	A	1/1990	Chang et al.
4,916,738	A	4/1990	Chandra et al.
5,048,085	A	9/1991	Abraham et al
5,150,407	A	9/1992	Chan
5,220,603	A	6/1993	Parker
5,265,159	A	11/1993	Kung
5,333,266	A	7/1994	Boaz et al.
5,388,255	A	2/1995	Pytlik
5,392,390	A	2/1995	Crozier
5,420,927	A	5/1995	Micali
5,425,102	A	6/1995	Moy
5,434,918	A	7/1995	Kung et al
5,483,596	A	1/1996	Rosenow et al.
5,491,752	A	2/1996	Kaufman et al.
5,495,533	A	2/1996	Linehan et al
5,510,777	A	4/1996	Pilc et al.
5,544,320	A	8/1996	Konrad

(Continued)

FOREIGN PATENT DOCUMENTS

CA	2191505	6/1997
CA	2210763	1/1999

(Continued)

OTHER PUBLICATIONS

US 5,373,559, 12/1994, Kaufman et al. (withdrawn)

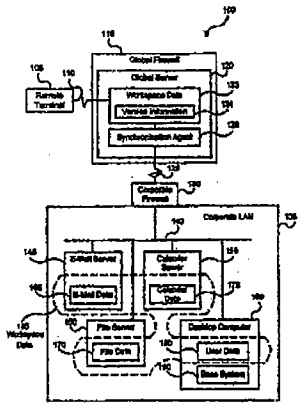
Rao, Venka & Aline, Mary, "Burrowing through firewalls," Dec 1996, pp 1–5.*

(Continued)

Primary Examiner—Alford W. Kindred

(57) **ABSTRACT**

A system includes a general synchronization module at the client site for operating within a first firewall and for examining first version information to determine whether a first workspace element has been modified. The system further includes a synchronization agent at a global server for operating outside the first firewall and for forwarding to the general synchronization module second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified. A synchronization-start module is maintained at the client site for operating within the first firewall and for securely initiating the general synchronization module and the synchronization agent when predetermined criteria have been satisfied. The system further includes means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information, and means for storing the preferred version at the first store and at the second store



U.S. PATENT DOCUMENTS

5,544,322	A	8/1996	Cheng et al.	
5,572,643	A	11/1996	Judson	
5,588,132	A	12/1996	Cardoza	
5,604,788	A	2/1997	Tett	
5,623,601	A	4/1997	Vu	
5,627,997	A	5/1997	Bray et al.	
5,632,011	A	5/1997	Landfield et al.	
5,634,053	A	5/1997	Noble et al	
5,644,354	A	7/1997	Thompson et al	
5,647,002	A	7/1997	Brunson	
5,652,884	A	7/1997	Palevich	
5,657,390	A	8/1997	Elgamal et al.	
5,664,207	A	9/1997	Crumpler et al.	
5,666,530	A	9/1997	Clark et al.	
5,675,782	A	10/1997	Montague et al.	
5,678,039	A	10/1997	Hinks et al	
5,680,542	A	10/1997	Mulchandani	
5,682,478	A	10/1997	Watson et al.	
5,684,951	A	11/1997	Goldman et al	
5,684,984	A	11/1997	Jones et al	
5,687,322	A	11/1997	Deaton	
5,701,400	A	12/1997	Amado	
5,706,427	A	1/1998	Tabuki	
5,710,922	A	1/1998	Alley et al	
5,713,019	A	1/1998	Keaten	
5,717,925	A	2/1998	Harper et al	
5,721,779	A	2/1998	Funk	
5,727,202	A	3/1998	Kucala	
5,729,735	A	3/1998	Meyering	
5,742,668	A	4/1998	Pepe et al.	
5,752,059	A	5/1998	Holleran et al	
5,752,246	A	5/1998	Rogers et al	
5,754,830	A	5/1998	Butts et al.	
5,758,150	A	5/1998	Bell et al.	
5,758,354	A	5/1998	Huang et al	
5,758,355	A	5/1998	Buchanan	
5,764,902	A	6/1998	Rothrock	
5,765,171	A	6/1998	Gehani et al	
5,768,510	A	6/1998	Gish	
5,784,463	A	7/1998	Chen et al.	
5,784,464	A	7/1998	Akiyama et al.	
5,787,172	A	7/1998	Arnold	
5,790,974	A	8/1998	Tognazzini	
5,794,252	A	8/1998	Mosher	
5,799,086	A	8/1998	Sudia	
5,802,530	A	9/1998	Van Hoff	
5,812,398	A	9/1998	Nielsen	
5,812,668	A	9/1998	Weber	
5,812,773	A	9/1998	Norin	
5,815,683	A	9/1998	Vogler	
5,818,935	A	10/1998	Maa	
5,828,840	A	10/1998	Cowan et al.	
5,832,483	A	11/1998	Barker	
5,835,087	A	11/1998	Herz et al.	
5,835,601	A	11/1998	Shimbo et al.	
5,845,282	A	12/1998	Alley et al.	
5,857,201	A *	1/1999	Wright et al.	707/104.1
5,862,325	A	1/1999	Reed et al	
5,862,346	A	1/1999	Kley et al.	
5,870,544	A	2/1999	Curtis	
5,870,759	A	2/1999	Bauer et al	
5,870,765	A	2/1999	Bauer et al.	
5,878,230	A	3/1999	Weber et al.	
5,909,689	A	6/1999	Van Ryzin	
5,924,103	A	7/1999	Ahmed et al	
5,928,329	A	7/1999	Clark et al	
5,943,676	A	8/1999	Boothby	
5,951,652	A	9/1999	Ingrassia, Jr. et al.	
5,961,590	A	10/1999	Mendez et al.	
5,966,714	A	10/1999	Huang et al	
5,968,131	A	10/1999	Mendez et al.	
5,974,238	A	10/1999	Chase, Jr.	
5,982,898	A	11/1999	Hsu et al.	
5,987,609	A	11/1999	Hasebe	
5,999,932	A	12/1999	Paul	
5,999,947	A	12/1999	Zollinger et al.	
6,006,017	A	12/1999	Joshi et al.	
6,006,274	A *	12/1999	Hawkins et al.	709/248
6,020,885	A	2/2000	Honda	
6,021,427	A	2/2000	Spagna et al.	
6,023,700	A	2/2000	Owens et al.	
6,023,708	A	2/2000	Mendez et al.	
6,034,621	A	3/2000	Kaufman	
6,052,735	A	4/2000	Ulrich et al.	
6,073,165	A	6/2000	Narasimhan et al.	
6,094,477	A	7/2000	Nada et al.	
6,108,691	A	8/2000	Lee et al.	
6,108,709	A	8/2000	Shinomura et al.	
6,118,856	A	9/2000	Paarsmarkt et al.	
6,125,281	A	9/2000	Wells et al.	
6,131,096	A	10/2000	Ng et al.	
6,131,116	A	10/2000	Riggins et al.	
6,138,146	A	10/2000	Moon et al	
6,151,606	A	11/2000	Mendez	
6,154,844	A	11/2000	Touboul et al.	
6,169,986	B1	1/2001	Bowman et al.	
6,182,118	B1	1/2001	Finney et al.	
6,212,529	B1	4/2001	Boothby et al.	
6,249,805	B1	6/2001	Fleming	
6,295,541	B1	9/2001	Bodnar et al.	
6,304,881	B1	10/2001	Halim et al.	
6,311,186	B1	10/2001	Melampy et al.	
6,324,542	B1	11/2001	Wright, Jr. et al.	
6,334,140	B1	12/2001	Kawamata	
6,343,313	B1	1/2002	Salesky et al.	
6,389,455	B1	5/2002	Fuisz	
6,438,583	B1	8/2002	McDowell et al	
6,446,090	B1	9/2002	Hart	
6,477,545	B1	11/2002	LaRue	
6,510,455	B1	1/2003	Chen et al	
6,564,218	B1	5/2003	Roth	
6,631,416	B2	10/2003	Bendinelli et al.	
6,697,942	B1	2/2004	L'Heureux et al	

FOREIGN PATENT DOCUMENTS

EP	0801478	10/1997
EP	0820028	1/1998
JP	02000003314	1/2000
WO	WO 96/17306	6/1996
WO	WO 97/04389	2/1997
WO	WO 97/24678	7/1997
WO	WO 97/35265	9/1997
WO	WO 99/05620	2/1999
WO	WO 99/45451	9/1999
WO	WO 00/22543	4/2000
WO	2002/0138599	9/2002
WO	2002/0174372	11/2002

OTHER PUBLICATIONS

IntelliLink Corporation, IteliLink for Windows Release 3.0, "User's Guide" 1994, Nashua, NH.

Lotus Development Corporation, Lotus Notes Release 4, "Application Developer's Guide" 1995, Cambridge, MA

Lotus Development Corporation, Lotus Notes Release 3.3 North American Server Edition, "Lotus Notes, the Groupware Standard" 1994, Cambridge, MA.

Sams Publishing, Dahl, Andrew, "Lotus Notes 4 Administrator's Survival Guide," 1996, Indianapolis, IN

Sams Publishing, Tamura, et al., "Lotus Notes 4 Unleashed," 1996, Indianapolis, IN

Advisor Publications—Lotus Notes Advisor, Pyle, Hugh, "The Notes Architecture," 1995.

Advisor Publications—Lotus Notes Advisor, Augun, Audry, "Integrating Lotus Notes with Enterprise Data," 1996.

Advisor Publications—Lotus Notes Advisor, Opyt, Barbara and Dale, Robert, "Use the Internet as Your Lotus Notes WAN," 1996.

Lotus Development Corporation, Lotus Notes Knowledge Base, "What is the Notes Replicator?" 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Knowledge Base, "Firewall Security Overview and How Firewalls Relate to Lotus Notes" 1996, Cambridge, MA.

Network Computing, Frenkel, Garry, "Pumping for Info: Notes and Database Integration," 1996.

IBM Corporation, Hawker et al, "Secrets to Running Lotus Notes: The Decisions No One Tells You How to Make," 1996, Research Triangle Park, NC.

Radosevich, Linda, "Users want unified mail directories," Computerworld, Aug. 30, 1993, p. 12.

Bruno, Charles, "Firm pushes E-mail limits beyond rivals'," Network World, Aug. 12, 1991, pp. 33, 53.

Sliwa, Carol and Cole, Barb, "MESA declares a groupware detente," Network World, Aug 5, 1996, p. 29.

Grosse, Eric, "Repository Mirroring," ACM Transactions on Mathematical Software, vol 21, No. 1, Mar, 1995, pp 89–97.

Schilit, Bill N. and Theimer, Marvin M., "Disseminating Active Map Information to Mobile Hosts," IEEE Network, Sep./Oct. 1994, pp. 22–32.

Levy, Eliezer and Silberschatz, Abraham, "Distributed File Systems: Concepts and Examples," ACM Computing Surveys, vol. 22, No. 4, Dec. 1990, pp. 321–374

Rymer, John R., "The Muddle in the Middle," Byte, Apr. 1996, pp 67–70.

Baum, David, "Intranet Politics and Technologies," Byte, May 1997, pp. 88A–88H.

Udell, Jon, "Push Me, Pull You," Byte, Sep. 1996, pp. 117–120.

Kador, John, "The Ultimate Middleware," Byte, Apr. 1996, pp. 79–83.

Brown, Mark, "Using Netscape 2," Que Corporation, 1995, pp. 885–907.

Kramer, Douglas, "The Java Platform," Javasoft, May 1996, pp 1–25.

Package java.applet; java.net, http://www-nrg.ee.lbl.gov/j4/apibook/javaf.htm, javam.htm, Apr. 22, 1996.

Servlet Tutorial, http://mech.postech.ac.kr/Java/java.sun.com/products/jeeves/currentrelease/doc/servlet_tutorial/servlet_tutorial.html.

Freier et al, entitled "The SSL Protocol, Version 3.0," Netscape Communications, dated Nov. 18, 1996, pp 1–59, Downloaded from URL=http://www.home.netscape.com/eng/ss13/draft302.txt.

Java Servlet Application Programming Interface White Paper, http://mech.postech.ac.kr/Java/java.sun.com/products/jeeves/CurrentRelease/doc/api.html, Aug. 1996, pp. 1–6.

Singhal, M., "Update transport: A new technique for update synchronization in replicated database systems," IEEE Transactions on Software Engineering, vol 16, No. 12, Dec 1990, pp 1325–1336.

Rao, H & Skarra, A., "A transparent service for synchronized replication across loosely-connected file systems," IEEE Transactions on Software Engineering, Apr. 1995, pp 110–117.

Crispin, M., "Internet Message Access Protocol—RFC 1730 Version 4," Dec. 1994, pp 1–52, http://www.faqs.org/rfcs/rfc1730.html.

Rao, Venkat & Aline, Mary, "Burrowing through firewalls," Dec. 1996, pp 1–5, http://java.sun.com/developer/technicalArticles/InnerWorkings/Burrowing/.

Brown, Margaret J., "The Visto Briefcase Pro Puts Your PIM On The Internet," Aug 13, 1998, 1 page, URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html

Tanenbaum, Andrew, "Computer Networks," Third Edition, Prentice–Hall, 1996, see entire document.

Knudsen, Jonathan, "Java Cryptography," O'Reilly, 1998, p 79–91, 112, 160.

Verisign Press Release, "Verisign Enhances Digital IDS to Enable Universal Website Login and One-step Registration," <www.verisign.com/press/product/isv.html>, especially 2nd paragraph

Crispin, M., "IMAP4 Compatibility With IMAP2BIS", Network Working Group RFC 2061, Dec. 1996, XP002154026, <URL:http://www.imc.org/rfc2061>, retrieved Nov. 28, 2000, p. 1, last paragraph.

"Configuration of a wireless–connected mobile computer," IBM Technical Disclosure Bulletin, US, IBM Corp., New York, vol. 38, No. 4, Apr. 1, 1995, pp. 15–17, XP000516057, ISSN: 0018–8689, the whole document.

Mason, Justin, "Tunneling over HTTP," Dec. 11, 1996, pp 1–2 http://www.netsys.com/firewalls/firewalls–9612/0488.html

Elgamal, Taher, "The Secure Sockets Layer Protocol (SSL)," agenda for the Danvers IETF meeting, Apr. 1995, pp 1–5—http://www.ietf.cnri.reston.va.us/proceedings/95apr/sec/cat.elgamal.slides.html

Gray, Terry, "Message Access Paradigms and Protocols," revised Sep. 28, 1995, pp 1–11 http://www.imap.org/imap.vs.pop.html.

Ouellette, Tim, "Data for everyone, bills for none?," Computerworld, Mar. 17, 1997, pp 43, 46.

Research Disclosure: "Provide Auto–Forwarding Based On Criteria Selected by the User"; Oct. 1, 1989; 1 page; No. 306; Kenneth Mason Publications; XP000085405; ISSN 0374–4353.

Chan, M.C. et al., "Application of Compaction Technique to Optimizing Wireless Email Transfer," IEEE Wireless Communications and Networking Conference 1999, p. 1535 Section A

Crispin, M., "Internet Message Access Protocol Version 4rev1," 1996 Section 2.3.1.1

Jaeger and Prakash, "Implementation of a Discretionary Access Control Model for Script–based Systems," IEEE Jun. 1995.

Adams, Charlotte, "Multilevel Secure Networking Charges Ahead," Federal Computer Week, Apr 12, 1993.

Kohl, John, "The Evolution of the Kerberos Authentication Service," 1991

Young et al., "Deniable Password Snatching: On the Possibility of Evasive Electronic Espionage," 1997, IEEE, pp. 224–235.

Abel, Amee, "E–mail anywhere," May 1998, Time, pp. 1–4.
Gray, Emmett, "Fetch–O–Matic by Emmett Gray," http://www.filmscouts.com/software/fomguide.asp, Version 2.5, 2001
Satyanarayanan, M., "Integrating Security in a Large Distributed System," ACM Transactions on Computer Systems, vol. 7, No. 3, Aug. 1989, pp. 247–280
Kistler, James J. and Satyanarayanan, M., "Disconnected Operation in the Coda File System," ACM Transactions on Computer Systems, vol. 10, No. 1, Feb. 1992, pp. 3–25.
Hills, Alex and Johnson, David B., "Wireless Data Network Infrastructure at Carnegie Mellon University," IEEE Personal Communications, 3(1), Feb. 1996.
Satyanarayanan, Mahadev, "Mobile Information Access," IEEE Personal Communications, Feb. 1996, pp. 26–33.
Satyanarayanan, Mahadev et al., "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Transactions on Computers, vol. 39, No. 4, Apr. 1990, pp. 447–459.
Satyanarayanan, Mahadev, "Scalable, Secure, and Highly Available Distributed File Access," Computer, May 1990, pp. 9–21.
Mummert, Lily B. et al., "Exploiting Weak Connectivity for Mobile File Access," SIGOPS '95, Dec. 1995, pp. 143–155.
Terry, Douglas B. et al., "Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System," SIGOPS '95, Dec. 1995, pp. 172–183
Demers, Alan et al., "The Bayou Architecture: Support for Data Sharing among Mobile Users," Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, Dec. 1994, pp. 2–7
Petersen, Karin et al., "Bayou: Replicated Database Services for World–wide Applications," Proceedings Seventh ACM SIGOPS European Workshop (EuroSIGOPS '96), Connemara, Ireland, Sep. 1996, pp. 275–280
Crocker, David H., "RFC822: Standard for ARPA Internet Text Messages," http://www.w3.org/Protocols/rfc822/.
Theisen, Tim, "AFS distributed filesystem FAQ (1/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.
Theisen, Tim, "AFS distributed filesystem FAQ (2/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.
Schubert, Eric, "Re: telnet/internet and . . . ," posted to comp.sys.hp.mpe newsgroup, Jun. 30, 1995.
Glazman, Daniel, "SOFTWARE: HHTPtool [sic] 1.1, a file transfer utility over HTTP using PUT and GET," posted to comp.infosystems.www.announce newsgroup, May 17, 1995
Glazman, Daniel, "SOFTWARE: HTTPtool v1.0 for Windows3.x, file transfer utility over HTTP," posted to comp.infosystems.www.announce newsgroup, Mar. 21, 1996.
Angus, Jeffrey G., "Sales force automation has a GoldMine," Computerworld, Oct. 7, 1996, p. 59.
Salamone, Salvatore, "Middle(ware) Management," Byte, Apr. 1996, pp. 71–76.
Nance, Barry, "Balance the Load with Transaction Server," Byte, Jun. 1997, pp. 81–84.
Francett, Barbara, "Replication on the Run," Software Magazine, Aug. 1996, pp. 63–66.
Darling, Charles B., EDA/SQL Loses a Little and Gains a Lot, Datamation, May 1, 1996, p. 12.
Varney, Sarah E., "Arm your salesforce with the Web," Datamation, Oct. 1996, pp. 72–74.
Fulcher, Jim, "Is it or isn't it?," Manufacturing Systems, Oct. 1996, pp. 56–61.
Lamb, John and Cusato, Tony, "LAN–Based Office for the Enterprise, A Case Study," Proceedings, 19th Conference on Local Computer Networks, Minneapolis, Minnesota, Oct. 2–5, 1994, pp. 440–447
Kawell Jr., Leonard et al., "Replicated Document Management In A Group Communication System," presented at the Second Conference on Computer–Supported Cooperative Work, Portland, Oregon, Sep. 26–28, 1988, as printed in Groupware: Software for Computer–Supported Cooperative Work, IEEE Computer Society Press, pp. 226–235.
Hong, Jack et al., "Personal Electronic Notebook with Sharing," Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Berkeley Springs, West Virginia, Apr. 20–22, 1995, pp. 88–94
Mace, Scott, "DataSync 2.0 enhances synchronization of data," InfoWorld, Jun. 6, 1994, p. 28
Mace, Scott, "Database replication products debut," InfoWorld, Nov. 15, 1993, p. 19.
Gryphon, Robert, "DataSync line matches up," InfoWorld, May 23, 1994, p. 161.
Patterson, John F. et al., "Notification Servers for Synchronous Groupware," Proceedings, ACM 1996 Conference on Computer Supported Cooperative Work, Boston, Massachusetts, Nov. 16–20, 1996, pp. 122–129.
Zhao, Jian and Hoppe, H. Ulrich, "Supporting Flexible Communication in Heterogeneous Multi–User Environments," 1994 IEEE, pp. 442–449.
Gong, Li, "Enclaves: Enabling Secure Collaboration Over the Internet," IEEE Journal on Selected Areas in Communications, vol. 15, No. 3, Apr. 1997, pp. 567–575.
Bruno, Lee, "Directory Services Tie It All Together," Data Communications, Mar. 1997, pp. 75–83.
Guy, Richard G. et al., "Implementation of the Ficus Replicated File System," Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, Jun. 11–15, 1990, pp. 63–70.
Page, Thomas W. Jr. et al., "Management of Replicated Volume Location Data in the Ficus Replicated Data System," Conference Proceedings, USENIX Summer 1991 Technical Conference, Nashville, Tennessee, Jun. 10–14, 1991, pp. 17–29
Reiher, Peter et al., "Resolving File Conflicts in the Ficus File System," Conference Proceedings, USENIX Summer 1994 Technical Conference, Boston, Massachusetts, Jun 6–10, 1994, pp. 183–195.
Walker, Bruce et al., "The LOCUS Distributed Operating System," Proceedings of the Ninth ACM Symposium on Operating Systems Principles, Bretton Woods, New Hampshire, Oct. 10–13, 1983, pp. 49–70.
Parker, D. Stott Jr. et al., "Detection of Mutual Inconsistency in Distributed Systems," IEEE Transactions on Software Engineering, vol. SE–9, No. 3, May 1983, pp. 240–246.
Ceri, Stefano et al., "The Case for Independent Updates," Second Workshop on the Management of Replicated Data, Monterey, California, Nov. 12–13, 1992, pp. 17–19.
Downing, Alan R. et al., "OSCAR: A System for WeakConsistency Replication," Proceedings, Workshop on the Management of Replicated Data, Houston, Texas, Nov. 8–9, 1990, pp. 26–30.

Siegel, Alex et al, "Deceit: A Flexible Distributed File System," Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, Jun. 11–15, 1990, pp. 51–61
Chutani, Sailesh, "The Episode File System," Conference Proceedings, USENIX Winter 1992 Technical Conference, San Francisco, California, Jan. 20–24, 1992, pp. 43–59.
Seltzer, Margo, "An Implementation of a Log–Structured File System for UNIX," Conference Proceedings, USENIX Winter 1993 Technical Conference, San Diego, California, Jan. 25–29, 1993, pp. 307–326.
Vahalia, Uresh, "Metadata Logging in an NFS Server," Conference Proceedings, USENIX 1995 Technical Conference on UNIX and Advanced Computing Systems, New Orleans, Louisiana, Jan. 16–20, 1995, pp. 265–276.
Lotus Development Corporation, InterNotes Web Publisher Release 4, "InterNotes Wed Publisher Guide" 1996, Cambridge, MA.
Lotus Development Corporation, Lotus Notes Release 4, "Database Manager's Guide" 1995, Cambridge, MA.
Lotus Development Corporation, Lotus Notes Release 4, "Administrator's Guide" 1995, Cambridge, MA.
Lotus Development Corporation, Lotus Notes Release 4, "Deployment Guide" 1995, Cambridge, MA
IBM Lotus Technical Library, Lotus Documentation, "Lotus Notes Internet Cookbook for Notes Release 3", Jan. 16, 1996, pp. 1–26 http://www–12 lotus com/ldd/doc/domino_notes/cookbook/cbookv4 nsf/e12503289bf7b3a385256
John Wiley & Sons, Inc.—Wiley Computer Publishing, Falkner, Mike, "How to Plan, Develop, and Implement Lotus Notes in Your Organization"1996, USA4.
McGraw–Hill, Lamb, John P and Lew, Peter W., "Lotus Notes Network Design for Notes Release 3 and 4", 1996, Quebecor–Fairfield, PA.
Lotus Notes, "Overview—What is Lotus NotesPump?", including "Notes Pump 1.0 Release Notes"
Lotus Development Corporation, Lotus Notes Release 3.1, The groupware standard, "Site and System Planning Guide" 1994, Cambridge, MA
Lotus Development Corporation, Lotus Notes Release 3.1. The groupwarestandard, "Administrator's Guide Server for NetWare, OS/2, and UNIX" 1994, Cambridge, MA.
IBM Lotus Technical Library, Lotus Documentation, "Lotus Notes Internet Cookbook for Notes Release 4", Date: Feb. 14, 1996, pp. 1–30 http://www–12 lotus.com/ldd/doc/domino_notes/cookbook/cbookv4 nsf/e12503289bf7b3a385256.
* cited by examiner

EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

ONLY THOSE PARAGRAPHS OF THE SPECIFICATION AFFECTED BY AMENDMENT ARE PRINTED HEREIN.

Column 4, lines 20–38:

An operating system **240** includes a program for controlling processing by the CPU **205**, and is typically stored in the data storage device **230** and loaded into the RAM **235** for execution A service engine **245** includes a program for performing a particular service such as maintaining an e-mail data base, a calendar data base, a bookmarks data base or another file data base, and may be also stored in the data storage device **230** and loaded into the RAM **235** for execution To perform a service, the service engine **245** operates on service data **250** (e.g., the e-mail data **165**, the file data **170**, the calendar data **175** or the user data **180**), which is typically stored in the data storage device **[250]** *230* The service data **250** includes version information **255** indicating the date and time of the last modification The service engine **245** operates to update the version information **255** whenever modifications are made It will be appreciated that the portion of memory in the data storage device **[250]** *230* which contains the service data **250** is referred to as the service "store."

Column 4, lines 44–59:

An operating system **340** includes a program for controlling processing by the CPU **305**, and is typically stored in the data storage device **330** and loaded into the RAM **335** for execution. A desktop service engine **345** (i.e , a particular service engine **245**, FIG. **2**) includes a service program for managing user data **180** (i.e., particular service data **250**, FIG. **2**) which includes version information **350** (i.e , particular version information **255**, FIG. **2**). The desktop service engine **345** may be also stored in the data storage devie **330** and loaded into the RAM **335** for execution. The user data **180** may be stored in the data storage **330** As stated above with reference to FIG. **1**, the base system **[1 90]** *190* operates to synchronize the workspace data **185** (which includes user data **180**) with the workspace data **123**. The base system **190** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution.

Column 5, lines 35–49:

The synchronization-start module **420** includes routines for determining when to initiate synchronization of workspace data **123** and workspace data **185**. For example, the synchronization-start module **420** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria The synchronization-start module **420** initiates data synchronization by instructing the general synchronization module **425** to begin execution of its routines. It will be appreciated that communications with synchronization agent **126** preferably initiate from within the

cooporate LAN **[1135]** *135*, because the typical cooporate firewall **130** prevents in-bound communications and allows out-bound communications.

Column 6, lines 15–27:

The content-based synchronization module **430** includes routines for reconciling two or more modified versions in workspace data **123**, **185** of the same workspace element. For example, if the original and the copy of a user workspace element have both been modified independently since the last synchronization, the **[**contentbased**]** *content-based* synchronization module **430** determines the appropriate responsive action. The content-based synchronization module **430** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores

Column 6, lines 28–41:

FIG **5** is a block diagram illustrating details of the synchronization agent **126**, which includes a communications module **505** (similar to the communications module **405** described above with reference to FIG. **4**) and a general synchronization module **510** (similar to the general synchronization module **425** described above also with reference to FIG **4**). The communications module **505** includes routines for compressing data, and routines for communicating via the communications channel **125** with the base system **190**. The communications module **505** may further include routines for establishing a secure communications channel through the global firewall **[126]** *115* and through the corporate firewall **130**.

Column 6, lines 42–56:

The general synchronization module **510** includes routines for comparing the version information **124** with the last synchronization signature **435**, and routines for forwarding to the general synchronization module **425** version information **124** determined to be modified The general synchronization module **510** may **[**either**]** maintain its own last synchronization signature **435** copy (not shown). Alternatively, the request to synchronize from the base system **190** may include a copy of the last synchronization signature **435**. The general synchronization module **510** further includes routines for receiving preferred versions of workspace data **185** workspace elements from the general synchronization module **425**, and routines for forwarding preferred versions of workspace data **123** workspace elements to the general synchronization module **425**.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

Claim **19** is cancelled.

Claims **1–2**, **6–8**, **10–11** and **21–25** are determined to be patentable as amended.

Claims **3–5**, **9**, **12–18** and **20**, dependent on an amended claim, are determined to be patentable.

1. A computer-based method comprising the steps of:

(a) *establishing a communications channel through a firewall using an HTTP port or an SSL port;*

(*b*) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within **[**a**]** *the* firewall has been modified;

(**[**b**]***c*) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store *on a smart phone* outside the firewall;

([c]*d*) initiating steps ([a]*b*) and ([b]*c*) from within the firewall *through the communications channel* when predetermined criteria have been satisfied;

([d]*e*) generating a preferred version from the first workspace element and from the copy based on the first and second examination results, *wherein if only one of the first workspace element and the copy has been modified, then the step of generating includes selecting the one as the preferred version*; and

([e]*f*) storing the preferred version at the first store and at the second store.

2. [The method of claim **1**] *A computer-based method comprising the steps of:*

(*a*) *generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;*

(*b*) *generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;*

(*c*) *initiating steps* (*a*) *and* (*b*) *from within the firewall when predetermined criteria have been satisfied,*

(*d*) *generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and*

(*e*) *storing the preferred version at the first store and at the second store;*

wherein the second store is on a global server outside the firewall and which is protected by a global firewall.

6. [The method of claim 1 further comprising,] *A computer-based method comprising the steps of:*

(*a*) *generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;*

(*b*) *generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;*

(*c*) *initiating steps* (*a*) *and* (*b*) *from within the firewall when predetermined criteria have been satisfied;*

(*d*) *generating a preferred version from the first workspace element and from the copy based on the first and second examination results;*

(*e*) *storing the preferred version at the first store and at the second store; and*

before generating the first examination results, the step of updating the first version information whenever the first workspace element is modified.

7 [The method of claim 1 further comprising,] *A computer based method comprising the steps of:*

(*a*) *generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;*

(*b*) *generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;*

(*c*) *initiating steps* (*a*) *and* (*b*) *from within the firewall when predetermined criteria have been satisfied;*

(*d*) *generating a preferred version from the first workspace element and from the copy based on the first and second examination results;*

(*e*) *storing the preferred version at the first store and at the second store; and*

before generating the second examination results, the step of updating the second version information whenever the copy is modified.

8 [The method of claim **1**] *A computer-based method comprising the steps of:*

(*a*) *generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;*

(*b*) *generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;*

(*c*) *initiating steps* (*a*) *and* (*b*) *from within the firewall when predetermined criteria have been satisfied;*

(*d*) *generating a preferred version from the first workspace element and from the copy based on the first and second examination results;*

(*e*) *storing the preferred version at the first store and at the second store; and*

wherein if only one of the first workspace element and the copy has been modified, then the step of generating includes selecting the one as the preferred version.

10 A system comprising:

a communications channel through a firewall comprising one of an HTTP port and an SSL port;

a general synchronization module for operating within [a] *the* first firewall and for examining first version information to determine whether a first workspace element *at a first store* has been modified;

a synchronization agent for operating outside the first firewall and for forwarding to the general synchronization module second version information which indicates whether an independently modifiable copy of the first workspace element *at a second store on a smart phone* has been modified;

a synchronization-start module for operating within the first firewall and for initiating the general synchronization module and the synchronization agent when predetermined criteria have been satisfied;

means for generating a preferred version from the first workspace element and from the copy by comparing the first version information and the second version information, *wherein if only one of the first workspace element and the copy has been modified, then the means for generating selects the one as the preferred version*; and

means for storing the preferred version at the first store and at the second store.

11. The system of claim **10** further comprising a communications module for communicating through the first firewall, *wherein the first firewall is positioned between a trusted network and the Internet*.

21 A system comprising:

first means for generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

second means for generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store *on a smart phone* outside the firewall;

means for updating the first version information whenever the first workspace element is modified or updating the second version information whenever the copy is modified;

means for initiating the first and second means from within the firewall when predetermined criteria have been satisfied;

means for generating a preferred version from the first workspace element and from the copy based on the first and second examination results; and

means for storing the preferred version at the first store and at the second store.

22. A computer-readable storage medium storing program code for causing a computer-*based system* to perform the steps of:

(a) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall *positioned between a trusted network and the Internet* has been modified;

(b) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store *on a smart phone* outside the firewall;

(c) initiating steps (a) and (b) from within the firewall *through an Internet communications channel* when predetermined criteria have been satisfied;

(d) generating a preferred version from the first workspace element and from the copy based on the first and second examination results, *wherein if only one of the first workspace element and the copy has been modified, then selecting the one as the preferred version*; and

(e) storing the preferred version at the first store and at the second store.

23. A computer-based method comprising the steps of:

(a) *establishing a secure communications channel through a firewall using an HTTP port or an SSL port;*

(*b*) generating first examination results from first version information which indicates whether a first workspace element stored at a first store within [a] *the* firewall has been modified;

(*c*) *before generating the first examination results, the step of updating the first version information whenever the first workspace element is modified;*

([b]*d*) generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

([c]*e*) initiating steps ([a]*b*) and ([b]*d*) from within the firewall when predetermined criteria have been satisfied;

([d]*f*) determining based on the first and second examination results that both the first workspace element and the copy have been modified; and

([e]*g*) storing both the first workspace element and the copy at the first store and at the second store, *wherein the second store comprises a smart phone*.

24. A system comprising:

first means for generating first examination results from first version information which indicates whether a first workspace element stored at a first store within a firewall has been modified;

second means for generating second examination results from second version information which indicates whether an independently-modifiable copy of the first workspace element has been modified, the copy being stored at a second store outside the firewall;

means for updating the first version information whenever the first workspace element is modified or updating the second version information whenever the copy is modified;

means for initiating the first and second means from within the firewall when predetermined criteria have been satisfied;

means for determining based on the first and second examination results that both the first workspace element and the copy have been modified; and

means for storing both the first [file] *workspace element* and the copy at the first store and at the second store, *wherein the second store comprises a smart phone*.

25. A system comprising:

a global server for operating outside a firewall and including memory for storing first workspace data and corresponding first version information; and

a synchronization agent for managing the first workspace data and the corresponding first version information and for communicating with remote clients; [and]

means for updating the first version information whenever the first workspace element is modified;

a remote client for operating within the firewall and including memory for storing second workspace data and corresponding second version information;

means for cooperating with the synchronization agent to synchronize the first workspace data with the second workspace data by examining the first version information and the second version information; and

a synchronization-start module for initiating workspace data synchronization between the global server and the remote client.

* * * * *

EXHIBIT C

US006151606A

United States Patent [19]

Mendez

[11] Patent Number: 6,151,606

[45] Date of Patent: Nov. 21, 2000

[54] **SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA**

[75] Inventor: **Daniel J. Mendez**, Menlo Park, Calif.

[73] Assignee: **Visto Corporation**, Mountain View, Calif

[21] Appl. No.: **09/008,354**

[22] Filed: **Jan. 16, 1998**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/201**; 707/8; 707/10; 707/202; 707/203

[58] **Field of Search** 707/8, 10, 202, 707/203, 506, 511; 709/103, 201, 204, 224, 228, 302, 303; 706/14, 45; 345/302, 340; 320/257, 463; 705/35; 395/500 32

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,831,582	5/1989	Miller et al	707/104
4,875,159	10/1989	Cary et al	364/200
4,897,781	1/1990	Chang	364/200
5,263,157	11/1993	Janis	707/9
5,386,564	1/1995	Shearer et al	707/500
5,392,390	2/1995	Crozier	395/161
5,572,643	11/1996	Judson	395/793
5,581,749	12/1996	Hossain et al	707/1
5,600,834	2/1997	Howard	395/617
5,613,012	3/1997	Hoffman et al	382/115
5,623,601	4/1997	Vu	395/187.01
5,627,658	5/1997	Connors et al	358/407
5,634,053	5/1997	Noble et al	707/4
5,647,002	7/1997	Brunson	380/49
5,652,884	7/1997	Palevich	395/651
5,666,530	9/1997	Clark et al	395/617
5,666,553	9/1997	Crozier	395/803
5,678,039	10/1997	Hinks et al	395/604
5,680,542	10/1997	Mulchandani	395/183.04
5,682,524	10/1997	Freund et al	395/605
5,684,990	11/1997	Boothby	707/203
5,687,322	11/1997	Deaton	705/14
5,701,400	12/1997	Amado	706/45
5,701,423	12/1997	Crozier	395/335
5,706,502	1/1998	Foley et al	707/10

(List continued on next page.)

OTHER PUBLICATIONS

Article by Bellovin et al., entitled: "Network Firewalls" Published by IEEE Communications Magazine Sep. 1994, pp. 50–57.

Article by Steffen Stempel, entitled: "IPAccess--An Internet Service Access System for Firewall Installations" Published by IEEE Communications Magazine Feb. 16, 1995, pp. 31–41.

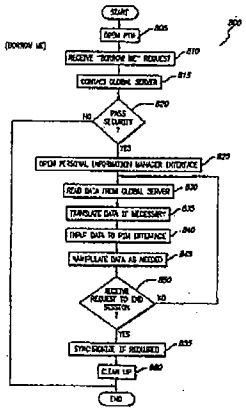
(List continued on next page.)

Primary Examiner—Thomas G. Black
Assistant Examiner—Diane D. Mizrahi
Attorney, Agent, or Firm—Graham & James LLP

[57] **ABSTRACT**

A system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader may translate the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager. Upon logout, a de-instantiator synchronizes the data with the global server and deletes the workspace data. The system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

21 Claims, 6 Drawing Sheets



U.S PATENT DOCUMENTS

5,710,918	1/1998	Lagarde et al.	395/610
5,713,019	1/1998	Keaten	395/610
5,715,403	2/1998	Stefik	705/44
5,717,925	2/1998	Harper et al.	395/613
5,721,908	2/1998	Lagarde et al.	395/610
5,721,914	2/1998	DeVries	707/104
5,729,735	3/1998	Meyering	395/610
5,745,360	4/1998	Leone et al.	364/140
5,757,916	5/1998	MacDoran et al.	380/25
5,758,150	5/1998	Bell et al.	395/610
5,758,354	5/1998	Huang et al.	707/201
5,758,355	5/1998	Buchanan	707/201
5,765,171	6/1998	Gehani et al.	707/203
5,778,346	7/1998	Frid-Nielsen et al.	395/208
5,790,425	8/1998	Wagle	364/551.01
5,790,790	8/1998	Smith et al.	395/200.36
5,799,318	8/1998	Cardinal et al.	707/104
5,832,483	11/1998	Barker	707/8
5,862,325	1/1999	Reed	395/200.31
5,870,759	2/1999	Bauer et al.	707/201
5,951,652	9/1999	Ingrassia et al.	709/248
5,966,714	10/1999	Huang et al.	707/201
5,999,947	12/1999	Zollinger et al.	707/203

OTHER PUBLICATIONS

Article by Braun et al, entitled: "Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server" Published by Elsevier Science B.V. 1995 pp. 37–51

Article by Nelson et al., entitled: "Security for Infinite Networks" Published by IEEE Communications Magazine on Aug. 22, 1995, pp 11–19.

Article by Greenwald et al, entitled: "Designing an Academic Firewall: Policy, Pratice, and Experience with SURF" Published by IEEE Communications Magazine on Feb. 22, 1996, pp 79–92.

Article by Kiuchi et al., entitled: "C–HTTP—The Development of a Secure, Closed HTTP–based Network on the Internet" Published by IEEE Proceedings of SNDSS on Feb. 22, 1996, pp. 64–75

Article by S. Cobb, entitled: "Establishing Firewall Policy" Published by National Computer Security Assn on Jun. 25–27, 1996, pp. 198–205.

Margaret J. Brown, "The Visto Briefcase Pro Puts Your PIM On The Internet", URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00 html, Aug. 13, 1998, 1 page

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome–Jul. 1997, URL=http://www bns it/emware/BookmarkTranslator–uk htm, pp. 1–4.

FIG. 1
(NETWORK SYSTEM)

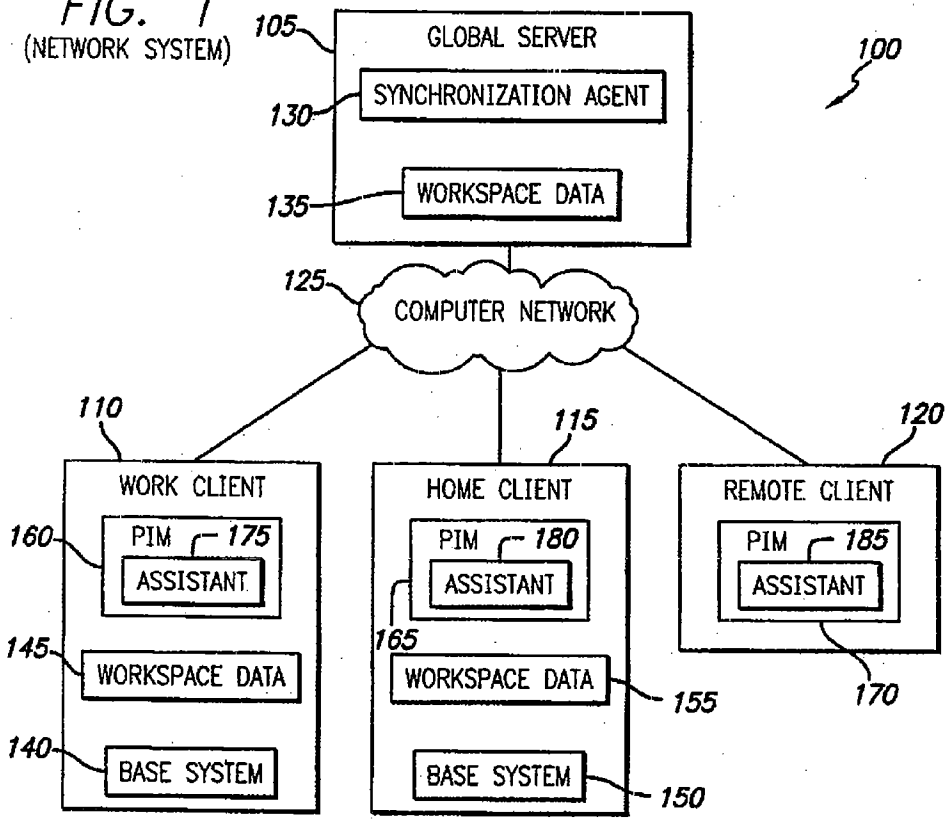
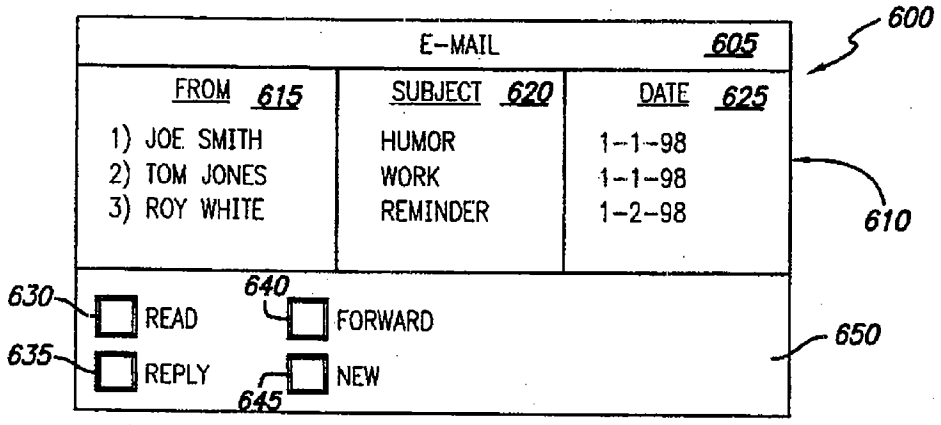


FIG. 6



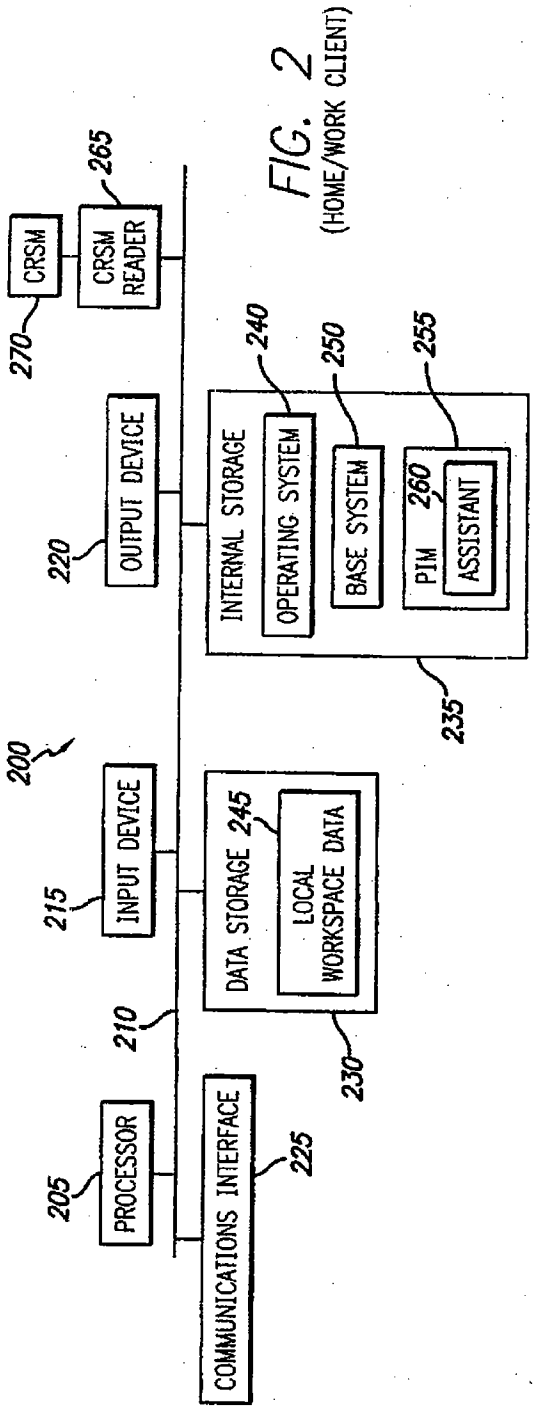


FIG. 2
(HOME/WORK CLIENT)

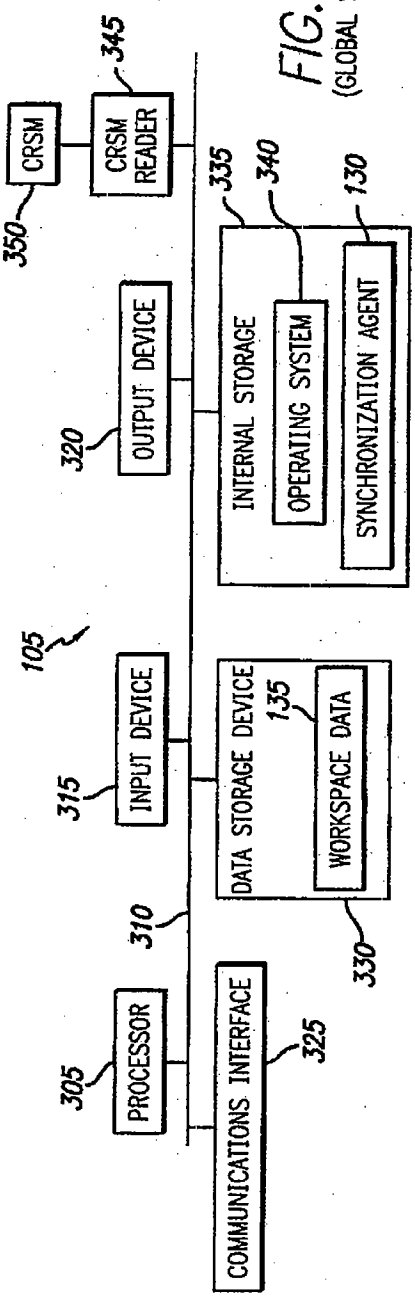


FIG. 3
(GLOBAL SERVER)

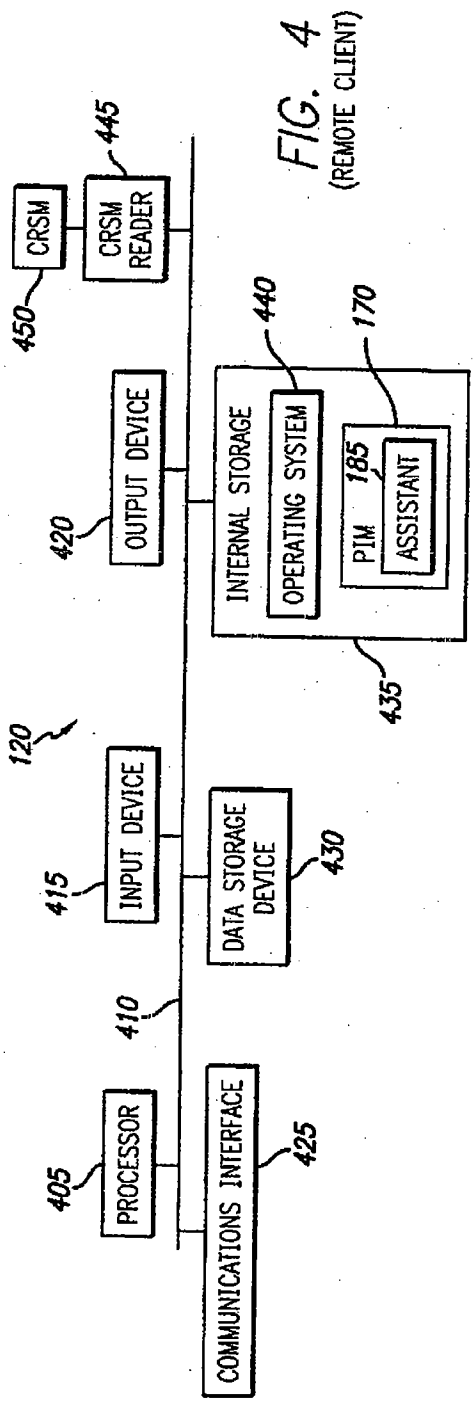


FIG. 4
(REMOTE CLIENT)

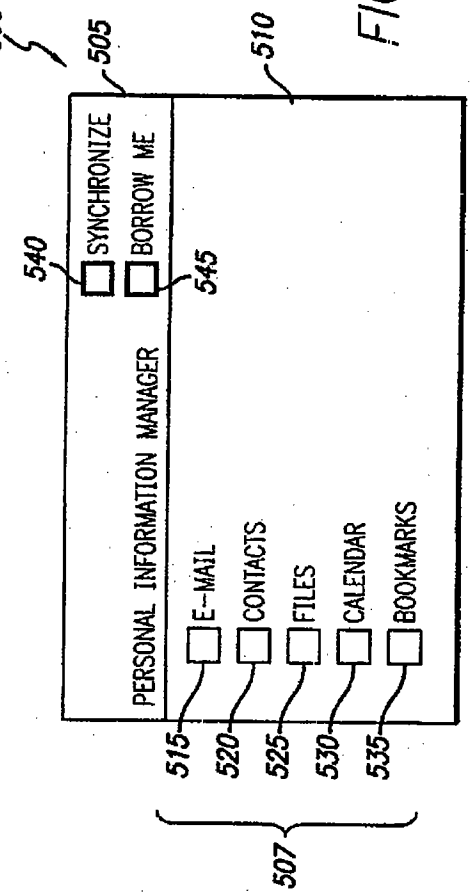


FIG. 5

FIG. 7

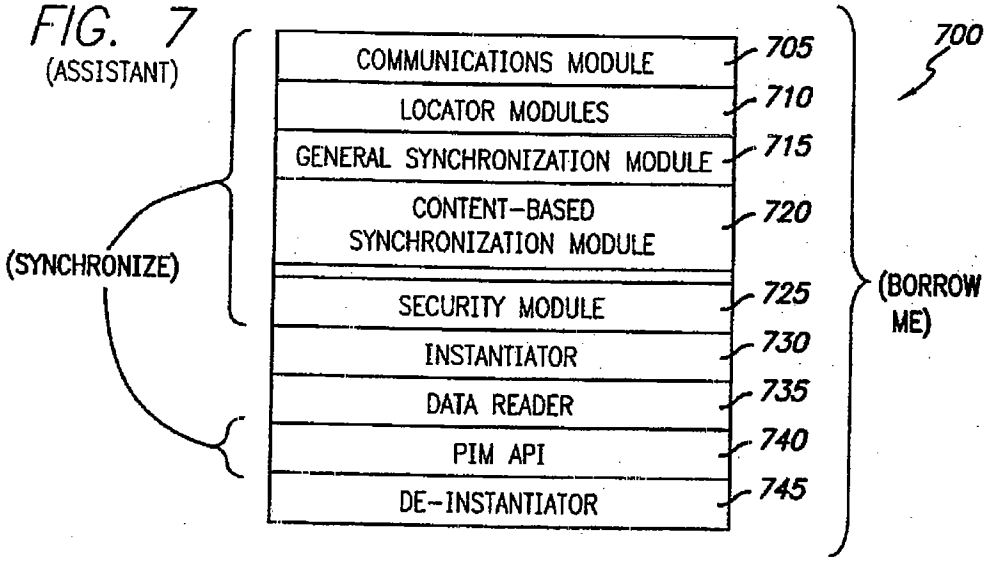
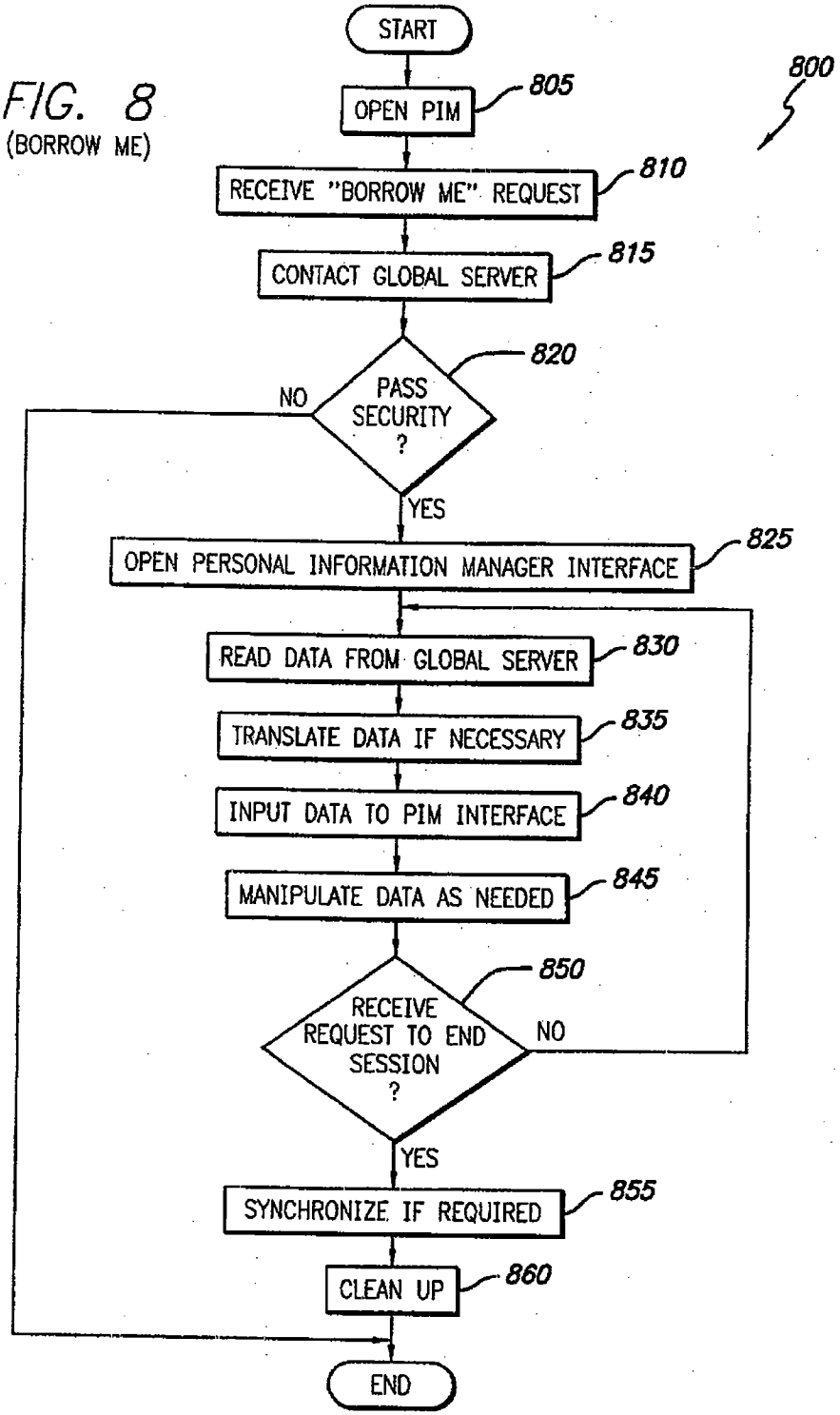

(ASSISTANT)
(SYNCHRONIZE)
COMMUNICATIONS MODULE
LOCATOR MODULES
GENERAL SYNCHRONIZATION MODULE
CONTENT-BASED SYNCHRONIZATION MODULE
SECURITY MODULE
INSTANTIATOR
DATA READER
PIM API
DE-INSTANTIATOR
705
710
715
720
725
730
735
740
745
700
(BORROW ME)

FIG. 8
(BORROW ME)



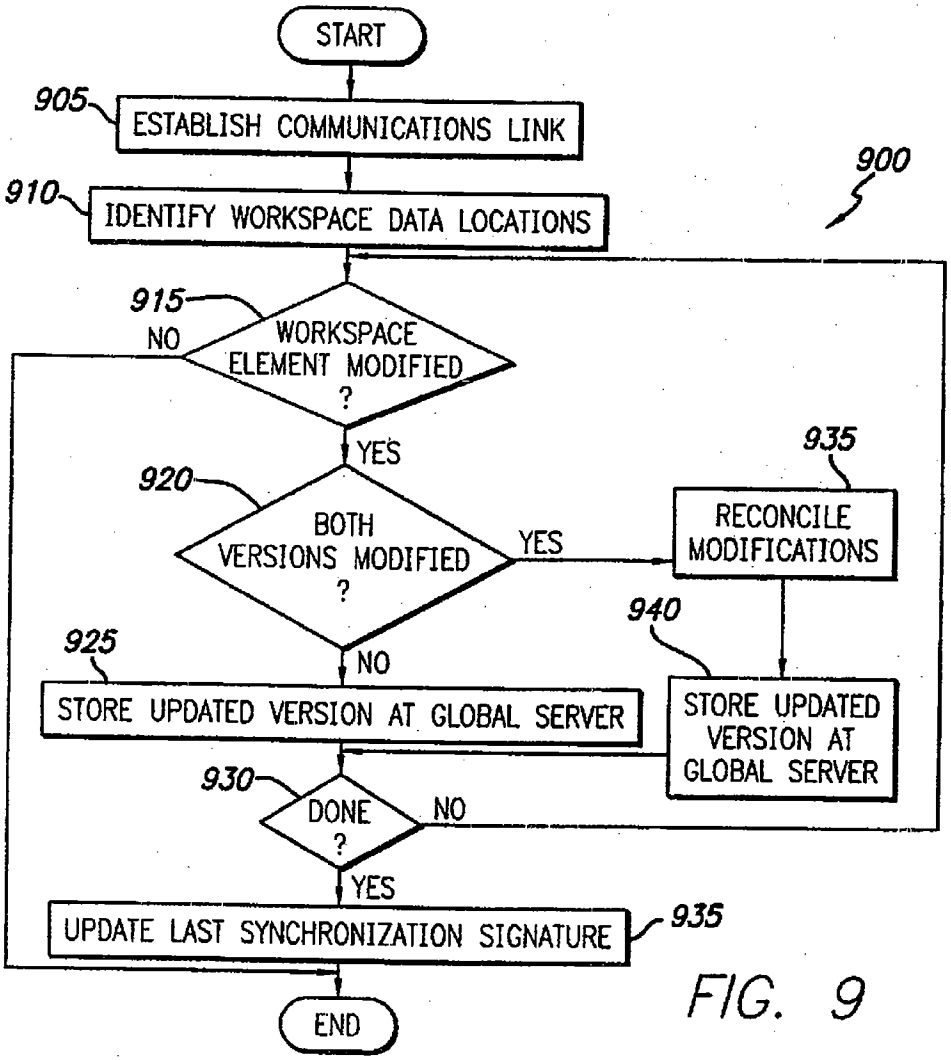

START
905
ESTABLISH COMMUNICATIONS LINK
910
IDENTIFY WORKSPACE DATA LOCATIONS
900
915
NO
WORKSPACE ELEMENT MODIFIED ?
YES
920
BOTH VERSIONS MODIFIED ?
YES
935
RECONCILE MODIFICATIONS
940
NO
925
STORE UPDATED VERSION AT GLOBAL SERVER
STORE UPDATED VERSION AT GLOBAL SERVER
930
DONE ?
NO
YES
UPDATE LAST SYNCHRONIZATION SIGNATURE
935
END

FIG. 9

SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA

PRIORITY REFERENCES

This application claims priority of and hereby incorporates by reference U.S. patent application Ser. No. 08/766,307 pending, entitled "System and Method for Globally Accessing Computer Services," filed on Dec. 13, 1996, by inventors Mark D. Riggins, et al; U.S. patent application Ser. No. 08/841,950 pending, entitled "System and Method for Enabling Secure Access to Services in a Computer Network, filed on Apr. 8, 1997, by inventor Mark D. Riggins; U.S. patent application Ser. No. 08/865,075, and now U.S. Pat. No. 6,023,708 entitled "System and Method for Using a Global Translator to Synchronize Workspace Elements Across a Network," filed on May 29, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/835,997 pending, entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," filed on Apr. 11, 1997, by inventors Daniel J. Mendez, et al.; U.S patent application Ser. No. 08/897,888 pending and now U.S Pat. No. 5,961,590, entitled "System and Method for Synchronizing Electronic Mail Across a Network," filed on Jul. 22, 1997, by inventors Daniel J. Mendez, et al.; U.S patent application Ser No. 08/899,277, entitled "System and Method for Using an Authentication Applet to Identify and Authenticate a User in a Computer Network," filed on Jul. 23, 1997, by inventor Mark D. Riggins; and U.S. patent application Ser No. 08/903,118 pending, entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network," filed on Jul. 30, 1997, by inventors Daniel J. Mendez, et al.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

This invention relates generally to computer networks, and more particularly provides a system and method for using a workspace data manager to access network data.

2. Description of the Background Art

Data accessibility and consistency are significant concerns for computer users. For example, when a roaming user, i.e., a user who travels to a remote location, needs to review or manipulate data such as an e-mail or prepared document, the roaming user must either carry the data to the remote location or access a workstation remotely. Maintaining a true copy of a database is a cumbersome process. Accordingly, system designers have developed an array of techniques for connecting a remote terminal across a computer network to the workstation storing the data.

To guarantee readability of the downloaded data, the user must carry a laptop computer containing all the applications needed to present and enable manipulation of the downloaded data, or find a network-connected computer that contains the needed application programs. Further, when maintaining multiple independently modifiable copies of particular data, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may already have resulted. The user must then spend more time reconciling the inconsistent versions

The problems of data accessibility and inconsistency are exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

SUMMARY OF THE INVENTION

The present invention provides a system for using a workspace data manager to access, manipulate and synchronize workspace data. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof

The system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader translates the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager. For example, data stored at the global server site in a canonical format may be translated to Organizer™, Outlook™ or other workspace element manager format. Upon logout, a de-instantiator initiates synchronization and deletes the data stored locally. It will be appreciated that the system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

The present invention further provides a method of using a workspace data manager to enable access, manipulate and synchronize workspace data. The method comprises the steps of downloading data from a remote site, requesting a workspace data manager to enable manipulation of the data and thereby create manipulated data, and synchronizing the manipulated data with the data stored at the remote site.

The system and method of the present invention advantageously enable the use of an integral interface, instead of using an interface for the synchronization software, an interface for the workspace data manager and an interface for the communication engine downloading the workspace data. Accordingly, the user need not become familiar with multiple interfaces. The user need only find a remote site that includes a workspace data manager that includes assistant-like functionality. Assistant-like functionality includes services for interfacing between the workspace data manager and the global server. Because the system and method substitute the global data for the local data, or create an instance for the global data, the system and method further advantageously enable a workspace data manager to provide

an interface for manipulating workspace data without compromising the local data.

Further, the system and method advantageously provide a simple graphical user interface for enabling borrowing of the workspace data manager and synchronization of manipulated data. The system and method also advantageously delete downloaded data and all interfaces from the local client, so that no traces are left on the local client for unprivileged users to review. Using the technology described in the applications incorporated by reference above, the system and method of the present invention further enable access and synchronization of data across different workspace data manager formats and across network firewalls

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a network system, in accordance with the present invention;

FIG. **2** is a block diagram illustrating details of the home or work client of FIG. **1**;

FIG. **3** is a block diagram illustrating details of the global server of FIG. **1**;

FIG **4** is a block diagram illustrating details of the remote client of FIG. **1**;

FIG. **5** is a block diagram illustrating details of an assistant of FIG. **1**;

FIG. **6** illustrates a personal information manager interface;

FIG. **7** illustrates a second personal information manager interface incorporating an assistant interface;

FIG. **8** is a flowchart illustrating a method of accessing network data from a remote site in accordance with the present invention; and

FIG. **9** is a flowchart illustrating a method of synchronizing network data from a remote site

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** is a block diagram illustrating a network system **100** for using a workspace data manager to access, manipulate and synchronize workspace data in accordance with the present invention. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof Although the network system **100** is described with reference to PIM's, one skilled in the art will recognize that the system **100** will work with any workspace data manager

Network system **100** includes a global server **105** coupled via a computer network **125** to a work client **110**, to a home client **115** and to a remote client **120**. The global server **105** includes a synchronization agent **130** and workspace data **135**. The work client **110** includes a base system **140** and workspace data **145**. The home client **115** includes a base system **150** and workspace data **155**.

Each of the base system **140** and the base system **150** cooperate with the synchronization agent **130** to synchronize workspace data **135**, workspace data **145** and workspace data **155** between the work client **110**, the home client **115** and the global server **105** Synchronization of workspace data **135**, **145** and **155** is described in detail in the patent applications incorporated by reference above. However, a brief example of synchronization is provided for completeness

First, the base system **140** on the work client **110** site negotiates a secure communications channel via any firewalls with the synchronization agent **130**, for example, using Secure Sockets Layer (SSL) technology. The base systems **140** examines version information and if necessary the content of a workspace to determine the most updated version. The most updated version is then stored at the client **110** site and at the global server **105** site. The base system **140** repeats these operations for all workspace elements selected for synchronization. Second, the base system **150** on the home client **115** site uses similar steps to synchronize its workspace data **155** with the workspace data **135** on the global server **105** site. Accordingly, the most updated versions of the workspace data **135**, **140** and **145** are stored at all three sites.

Each of the work client **110**, the home client **115** and the remote client **120** includes a respective workspace data manager, e.g., a Personal Information Manager (PIM) **160**, **165** and **170** such as Outlook™ 98 developed by Microsoft Corporation, Organizer 97 developed by Lotus Development Corporation or Sidekick 98 developed by Starfish Software. Each PIM **160**, **165** and **170** includes an assistant **175**, **180** and **185** that adds data access and synchronization functions to the PIM **160**, **165** and **170**. Accordingly, a user can transparently use an assistant **175**, **180** or **185** via a PIM **160**, **165** or **170** to access workspace data **135** from the global server **105**, to present and enable manipulation of downloaded workspace data **135**, and to synchronize manipulated downloaded data **135** with the workspace data **135** stored on the global server **105**. Components and operations of the assistant **175**, **180** or **185** are described in detail with reference to FIGS **7–9**.

FIG **2** is a block diagram illustrating details of a data-synchronizing client **200**, in a generic embodiment which exemplifies each of the work client **110** and the home client **115**. The client **200** includes a processor **205**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **210**. The client **200** further includes an input device **215** such as a keyboard and mouse, an output device **220** such as a Cathode Ray Tube (CRT) display, data storage **230** such as a magnetic disk, and internal storage **235** such as Random-Access Memory (RAM), each coupled to the communications channel **210** A communications interface **225** couples the communications channel **210** to the computer network **125**.

An operating system **240** controls processing by processor **205**, and is typically stored in data storage **230** and loaded into internal storage **235** (as illustrated) for execution. A base system **250**, which cooperates with the synchronization agent **130** for synchronizing local workspace data **245** with workspace data **135**, also may be stored in data storage **230** and loaded into internal storage **235** (as illustrated) for execution by processor **205**. The local workspace data **245** exemplifies workspace data **145** or workspace data **150**, and may be stored in data storage **230**.

A PIM **255** includes an assistant **260**, which enables a user to download workspace data **135** from the global server **105**, and to use the PIM **255** for displaying and manipulating the workspace data **135**. The assistant **260** further enables the PIM **255** to synchronize the manipulated data **135** with the

workspace data **135** on the global server **105**. The PIM **255** exemplifies each of the PIM **160** on the work client **110** and the PIM **165** on the home client **115** The assistant **260** exemplifies each of the assistant **175** on the work client **110** and the assistant **180** on the home client **115** The PIM **255** may be stored in data storage **230**, and loaded into internal storage **235** (as illustrated) for execution by the processor **205**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a computer-readable storage medium (CRSM) reader **265** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **210** for reading a computer-readable storage medium (CRSM) **270** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **265**.

FIG. **3** is a block diagram illustrating details of the global server **105**. The global server **105** includes a processor **305**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **310**. The global server **105** further includes an input device **315** such as a keyboard and mouse, an output device **320** such as a CRI display, data storage **325** such as a magnetic disk, and internal storage **330** such as RAM, each coupled to the communications channel **310** A communications interface **325** couples the communications channel **310** to the computer network **125**

An operating system **340** controls processing by processor **305**, and is typically stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution The synchronization agent **130**, which cooperates with the base system **250** (FIG. **2**) for synchronizing local workspace data **245** with workspace data **135**, also may be stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**. The workspace data **135** may be stored in data storage **230**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **345** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **350** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc Accordingly, the system **100** may receive programs and data via the CRSM reader **345**.

FIG. **4** is a block diagram illustrating details of the remote client **120**. The client **120** includes a processor **405**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **410**. The client **120** further includes an input device **415** such as a keyboard and mouse, an output device **420** such as a CRI display, data storage **425** such as a magnetic disk, and internal storage **430** such as RAM, each coupled to the communications channel **410**. A communications interface **425** couples the communications channel **410** to the computer network **125**.

An operating system **440** controls processing by processor **405**, and is typically stored in data storage **430** and loaded into internal storage **435** (as illustrated) for execution The PIM **170** and assistant **185** may be stored in data storage **430**, and loaded into internal storage **435** (as illustrated) for execution by the processor **405**

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **445** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **450** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **445**.

FIG **5** is a block diagram illustrating a PIM interface **500**, which includes a header **505** and a selection window **510**

The header **505** includes a synchronize button **540** and a "borrow me" button **545**, which are presented by the assistant **175**, **180** or **185** incorporated in the PIM **160**, **165** or **170** Invoking the synchronize button **540** causes the assistant **175**, **180** or **185** to enable synchronization of data entered into the PIM **160**, **165** or **170** with the workspace data **135** on the global server **135** The synchronize button **540** may enable the user to configure a preference file that indicates when automatic synchronization is to initiate and may also enable a user to effect manual synchronization.

The "borrow me" button **545** enables a user to use a PIM **160**, **165** or **170** for viewing and manipulating workspace data **135** downloaded from the global server **105**. That is, invoking the "borrow me" button **545** causes the corresponding assistant **175**, **180** or **185** to communicate with the global server **105**, to provide user identification and authentication information to the global server **105**, to download workspace data **135** from the global server **105**, to display and enable manipulation of the downloaded data **135** using the PIM interface **500**, and to synchronize the manipulated downloaded data **135** upon logout. Since the PIM interface **500** is provided by the pre-existing PIM, the assistant **175**, **180** or **185** need not provide its own data interface. Only a single interface is needed.

It will be appreciated that upon logout, the base systems **140** and **150** will cooperate with the synchronization agent **130** to synchronize automatically the workspace data **135** on the global server **105** with the workspace data **145** and **155**. Accordingly, the user always has access to the most updated versions of workspace data from any site that executes a PIM **160**, **165** or **170** having an assistant **175**, **180** or **185** embodied therein.

It will be appreciated that the synchronize button **540** is most helpful to the work client **110** and the home client **115**, since typically the work client **110** and home client **115** will set the preference file to configure automatic synchronization. Synchronization of the manipulated workspace data **135** at the remote client **120** will most often be effected through the automatic logout procedures of the "borrow me" button. Logout is described in greater detail with reference to the Outlook™ and Lotus Organizer examples shown and described below with reference to FIG. **7**. Accordingly, the borrow me button **545** is most helpful to the remote client **120**.

The selection window **510** provides a list of buttons **507**, wherein each button **507** corresponds to a set of workspace elements, e.g., e-mails **515**, contacts **520**, files **525**, calendar data **530** and bookmarks **535**. A mouse-down on a virtual button **507** causes the selection of a corresponding workspace element set and the selection of a corresponding user interface for displaying and enabling manipulation of the workspace elements included in the set. For example, selection of button **515** selects the e-mail set, and selects a corresponding user interface for displaying, writing, forwarding, etc. e-mails. Selecting a button **507** causes the assistant **175**, **180** or **185** to download the corresponding workspace data **135**, and causes the PIM **160**, **165** or **170** to display and enable manipulation of the downloaded data **135** on a workspace element set interface (shown and described with reference to FIG. **6**)

FIG. **6** illustrates an example e-mail workspace element set user interface **600** (commonly referred to as the "In-Box") for displaying received e-mails. The user interface **600** includes a header **605**, an e-mail list window **610** and a manipulation command window **650**.

The header **605** lists the name of the workspace element set, namely, "E-Mail." The e-mail list window **610** comprises three columns, including an origin column **615** which provides the origin of each e-mail, a subject column **620** which provides the subject of each e-mail, and a date column **625** which provides the date each e-mail was received. The e-mail list window **610** may display e-mails stored in a local e-mail database (not shown), e-mails stored in the e-mail server (not shown) or e-mails downloaded from the global server **105**. The e-mails shown include a first e-mail from Joe Smith, a second e-mail from Tom Jones, and a third e-mail from Roy White. If the user depressed the "borrow me" button **545** shown in FIG **5**, then the e-mail list displayed would be the e-mails stored and downloaded from the global server **105**.

The manipulation window **650** includes available functions such as the conventional e-mail read function **630**, e-mail reply function **635**, e-mail forward function **640** and new e-mail write function **645**. It will be appreciated that the columns and functions will vary based on the PIM

FIG. **7** is a block diagram illustrating details of a generic assistant **700**, which exemplifies each of the assistant **175**, **180** and **185**. The generic assistant **700** includes a communications module **705**, locator modules **710**, a general synchronization module **715**, a content-based synchronization module **720**, a security module **725**, an instantiator **730**, a data reader **735**, a PIM Application Program Interface (API) **740** and a de-instantiator **745**. The synchronization function of the assistant **700** uses the communications module **705**, the locator modules **710**, the general synchronization module **715**, the content-based synchronization module **720**, the security module **725** and the PIM API **740**. The "borrow me" function of the assistant **700** uses the communications module **705**, the locator modules **710**, the security module **725**, the instantiator **730**, the data reader **735**, the PIM API **740** and the de-instantiator **745**.

The communications module **705** includes routines for compressing and decompressing data, and routines for communicating with the synchronization agent **130**. The communications module **705** may apply Secure Socket Layer (SSL) technology to establish a secure communication channel. Examples of communications modules **705** may include TCP/IP stacks or the AppleTalk protocol

The locator modules **710** include routines for identifying the memory locations of the workspace elements in the workspace data **135**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query the user. Accordingly, the locator modules **710** determine the memory addresses of the workspace elements in e-mail workspace data **135**, in file workspace data **135**, in calendar workspace data **135**, etc

The general synchronization module **715** examines the workspace data **135** on the global server **105** to determine whether it had been modified while the user manipulated the data on the client **110**, **115** or **120**. Further, the general synchronization module **715** determines whether the user manipulated any data on the client **110**, **115** or **120**. If the general synchronization module **715** determines that only the data on the client **110**, **115** or **120** was manipulated, then the general synchronization module **715** computes and sends the changes to the synchronization agent **130** of the global server **105**. The general synchronization module **715** is initiated when the synchronization button **540** is depressed and during the logout procedures of the "borrow me" function.

The synchronization agent **130** then updates a last synchronization signature to indicate to all base systems **140** and **150** that synchronization with workspace data **145** and synchronization with workspace data **155** are needed. If the general synchronization module **715** determines that changes were made only to the workspace data **135** on the global server **105**, then the general synchronization module **715** instructs the synchronization agent **130** to compute and transmit the changes made to the client **110**, **115** or **120** at the client's request. The client **110** or **120** then updates its information. It will be appreciated that sending only the changes reduces processor load and increases transmission line efficiency, although alternatively an entire manipulated workspace element can be sent to the global server **105**.

If the general synchronization module **715** determines that the workspace data **135** on the global server **105** has been modified since download, and that the data on the client **110**, **115** or **120** has been modified, then the general synchronization module **715** instructs the content-based synchronization module **720** to perform its duties. The content-based synchronization module **720** includes routines for reconciling two or more modified versions of a workspace element. The content-based synchronization module **720** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores

The security module **725** includes routines for obtaining user identification and authentication using such techniques as obtaining login and password information, obtaining a response to a challenge, obtaining a public key certificate, etc. The security module **725** performs identification and authentication techniques to confirm authorization by the user to access the workspace data **135** stored on the global server **105**. It will be appreciated that authorization may be granted only to portion of the workspace data **135** that belongs to the user.

The instantiator **730** is an application program interface **730** that creates a window for displaying and enabling manipulation of the workspace data **135** downloaded from the global server **105**. In an object-oriented environment, the instantiator **730** may create a new instance for the workspace data **135**. Alternatively, the instantiator **730** may store the

local data to a buffer (not shown) and use the current interface to display and enable manipulation of the workspace data **135**.

The data reader **735** communicates with the synchronization agent **130** at the global server **105**, and retrieves the workspace data **135** requested. For example, if the user depresses the "borrow me" button **545** (FIG **5**) and depresses the e-mail button **515**, then the data reader **735** retrieves the e-mail workspace elements of the workspace data **135**, and delivers them to the PIM API **740**.

The PIM API **740** translates and transfers the workspace data **135** received from the global server **105** to the PIM **160**, **165** or **170** for display and enabling manipulation thereto. The PIM API **740** further translates and transfers the workspace data manipulated on the client **110**, **115** or **120** from the PIM **160**, **165** or **170** back to the global server **105**.

The de-instantiator **745** returns the PIM **160**, **165** or **170** to the state before the user selected the "borrow me" button **545**. The user may initiate operations of the de-instantiator **745** by depressing an "unborrow me" button (not shown) that is presented after selection of the "borrow me" button **545**. The de-instantiator **745** deletes any instance created by the instantiator **730**, deletes all workspace data **135** and data created by the user on the client **110**, **115** or **120** and automatically initiated synchronization of any manipulated downloaded data **135** with the workspace data **135** stored at the global server **105**.

Operations of the instantiator **730**, the data reader **735**, the PIM API **740** and the de-instantiator **745** are described in greater detail with reference to the following examples:

OUTLOOK EXAMPLE

Action	Global Data	Local Data
standby	—	local data → pst^{local}
button depressed	—	pst^{local}
enter login/password	—	pst^{local}
authenticate	—	pst^{local}
send global data	global data → pst^{local}	local data → pst^{buffer}
manipulate data	global data → global data 2	pst^{buffer}
logout	1) Compute Δglobal data 2) Synchronize Δglobal data with global server 3) Delete global data 2	
	4)	local data → pst^{local}

As illustrated by the Outlook™ example above, during standby, the PIM **160**, **165** or **170** stores the local data on the client **110**, **115** or **120** in a personal folder store pst^{local} The user then depresses the "borrow me" button **545** The security module **725** requests the user to enter a login and password, which the global server **105** authenticates. During these steps, it will be appreciated that the local data remains stored in pst^{local}. Upon user identification and authentication, the global server **105** sends the workspace data **135** (global data) to the requesting client **110**, **115** or **120**. The instantiator **730** on the client **110**, **115** or **120** transfers the local data from pst^{local} to a buffer pst^{buffer}, and stores the received global data into pst^{local}. The data reader **745** and PIM API **740** enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, for example, after an "unborrow me" button (not shown) is depressed, the global data 2 is synchronized with the workspace data **135**. Namely, the general synchronization module **715** determines the changes made (Δglobal data), and synchronizes Δglobal data with the workspace data **135**. The de-instantiator **745** deletes global data 2 and Δglobal data, and returns the local data to pst^{local}.

LOTUS ORGANIZER EXAMPLE

Action	Global Data	Local Data
standby	—	local.org
button	—	local.org
enter login/password	—	local.org
authenticate	new instance	local.org
send global data	open with global.org	local.org
manipulate data	global.org → $global.org^2$	local.org
logout	1) compute Δglobal.org 2) Synchronize Δglobal.org with global server 3) delete $global.org^2$	local.org

As illustrated by the Lotus Organizer example above, during standby, the PIM **160**, **165** or **170** stores the local data on the client **110**, **115** or **120** in local.org. The user then depresses the "borrow me" button **545**. The security module **725** requests the user to enter a login and password, which the global server **105** authenticates. During these steps, it will be appreciated that the local data remains stored in local.org. Upon user identification and authentication, the global server **105** sends the workspace data **135** (global data) to the requesting client **110**, **115** or **120**. The instantiator **730** on the client **110**, **115** or **120** creates a new instance, e.g., a new window, of PIM API **740** and stores the received global data into another file, i.e , global.org. The data reader **745** and PIM API **740** enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, the global data 2 is synchronized with the workspace data **135**. Namely, the general synchronization module **715** determines the changes made (Δglobal data), and synchronizes Δglobal data with the workspace data **135**. The de-instantiator **745** deletes global data 2, Δglobal data and global.org

FIG **8** is a flowchart illustrating a method **800** of accessing data remotely in accordance with the present invention. The method **800** begins with the processor **405** in step **805** opening the PIM **160**, **165** or **170** per user request, and the PIM **160**, **165** or **170** opening a PIM interface **500** (FIG. **5**). The PIM **160**, **165** or **170** in step **810** receives a "borrow me" request from the user, i.e., the user depresses the "borrow me" button **545**. The PIM API **740** in step **815** recognizes the request, and instructs the communications module **705** to create a communications link with the global server **105**.

The security module **725** in step **820** requests and transmits identification and authentication information such as login and password information from the user to the global server **105** for examination. If the global server **105** fails to identify or authenticate the user, then the method **800** ends. Otherwise, the instantiator **730** in step **825** opens a PIM interface **500** to display and enable manipulation of the workspace data **135** downloaded from the global server **105**. The data reader **735** in step **830** reads the workspace data **135** downloaded from the global server **105**, and in step **835** translates the data to the appropriate format if necessary. That is, the data reader **735** translates the workspace data **135** from the format implemented by the global server **105** to the format implemented by the PIM **160**, **165** or **170**. The PIM API **740** in step **840** passes the translated workspace data **135** to the PIM interfaces **500** and **600**.

The PIM **160**, **165** or **170** enables the user in step **845** to manipulate the workspace data **135** as necessary. Manipu-

lation includes adding new data, deleting workspace data **135**, editing workspace data **135**, etc. For example, the user can depress the e-mail button **515** in interface **500** to select, review and manipulate e-mail in interface **600**, and then can depress the calendar button **530** in interface **500** to select, review ad manipulate calendar information (not shown) in an interface similar to the e-mail interface **600** In step **850**, the PIM API **740** waits to receive an "end session" request. Until an "end session" request is received, the method **800** returns to step **830** to enable continued data review and manipulation.

Upon receiving an "end session" or "unborrow me" request, the de-instantiator **745** initiates the general synchronization module **715** in step **855** to synchronize the manipulated workspace data on the client **110, 115** or **120** with the workspace data **135** on the global server **105**, if required. Synchronization is described in greater detail with reference to FIG **9**. The de-instantiator **745** in step **860** deletes the workspace data on the client **110, 115** or **120**, and deletes all records of the matter. Method **800** then ends.

FIG. **9** is a flowchart illustrating a method **900** for synchronizing workspace data in a computer network **100**. Method **900** begins with the communications module **705** in step **905** establishing a communications link with the synchronization agent **130** of the global server **105**. The locator modules **710** in step **910** identify the memory locations of the workspace elements in the workspace data **135**. It will be appreciated that workspace element memory location identification may be implemented using intelligent software or dialogue boxes

The general synchronization module **715** in step **915** compares version information (not shown) for each workspace element in the workspace data (on the client **110, 115** or **120** and on the global server **105**) against a last synchronization signature to determine which workspace elements have been modified. In this embodiment, a workspace element may have been modified if the date and time of the last modification is after the date and time of the downloading.

If the general synchronization module **715** locates no modified workspace elements in the workspace data on the client **110, 115** or **120**, then the method **900** ends Otherwise, the general synchronization module in step **920** determines whether the version of the same workspace element of the workspace data **135** on the global server **105** has been modified since the data **135** was downloaded.

If only the version on the client **110, 115** or **120** has been modified, then the general synchronization module **715** in step **925** stores the updated version of the workspace element at the global server **105**. To store the updated version on the global server **105**, the general synchronization module **715** may compute the changes made and forward the changes to the synchronization agent **130**. The synchronization agent **130** enters the changes into the global server **105** version. The general synchronization module **715** in step **930** determines whether all workspace elements downloaded to the client **110, 115** or **120** have been examined. If not, then method **900** returns to step **915**. Otherwise, the synchronization agent **130** in step **935** updates the last synchronization signature, and method **900** ends. Updating the last synchronization signature will instruct the base systems **140** and **150** to synchronize the workspace data **145** and **155** with the workspace data **135** on the global server **105**, as described in the patent applications incorporated by reference above.

If the general synchronization module **715** in step **920** determines that both the version on the client **110, 115** or **120** and the version on the global server **105** have been modified, then the general synchronization module in step **935** instructs the content-based synchronization module **729** to reconcile the modified versions Reconciliation may include requesting instructions from the user, or performing based on pre-selected preferences responsive actions such as storing both versions at the global server **105**. The general synchronization module **715** in step **940** stores the preferred version on the global server **105**. Method **900** then proceeds to step **930**.

The foregoing description of the preferred embodiments of the present invention is by way of example only, and other variations and modifications of the above-described embodiments and methods are possible in light of the foregoing teaching. Although the network sites are being described as separate and distinct sites, one skilled in the art will recognize that these sites may be a part of an integral site, may each include portions of multiple sites, or may include combinations of single and multiple sites. Further, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. Connections may be wired, wireless, modem, etc The embodiments described herein are not intended to be exhaustive or limiting. The present invention is limited only by the following claims.

What is claimed is:

1. A computer-based method, comprising the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote being connected via a network to untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data

2. The method of claim **1**, further comprising the step of requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

3. The method of claim **2**, further comprising the steps of using the workspace data manager to manipulate the dow data, thereby creating manipulated data, using the workspace data manager interface to request synchronization, and synchronizing the manipulated data with the data at the remote site.

4. The method of claim **3**, wherein the data at the remote site has not been modified after the step of downloading and before the step of synchronizing and therefore includes the downloaded data.

5 The method of claim **3**, wherein the data at the remote site has been modified after the step of downloading and before the step of synchronizing, and therefore is different than the downloaded data

6. The method of claim **2**, wherein the workspace data manager provides an interface by creating an instance.

7. The method of claim **2**, wherein the workspace data manager provides an interface by providing access to its only interface.

8. The method of claim **1**, further comprising the step of translating the downloaded data from the format used by the remote site and the format used by the workspace data manager

9. The method of claim 1, further comprising the step of deleting the workspace data manager interface after it is no longer needed

10. A system on an untrusted client site, comprising:

a communications module for download data from a remote site, the remote site being connected via a network to the untrusted client site;

program code for placing the downloaded data in temporary storage on the untrusted client site;

an application program interface coupled to the communications module for communicating with a workspace data manager to present the downloaded data; and

program code coupled to the application program interface for automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data

11. The system of claim 10, further comprising an instantiator for requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

12. The system of claim 11, wherein the workspace manager enables manipulation of the downloaded data to create manipulated data and the data manipulation interface enables a request to synchronize the data, and further comprising a synchronization module coupled to the communications module for enabling synchronization of the manipulated data with the data at the remote site.

13. The system of claim 12, wherein the data stored at the remote site has not been modified and therefore includes the downloaded data.

14. The system of claim 12, wherein the data stored at the remote site has been modified, and therefore is different than the downloaded data

15. The system of claim 14, further comprising a content-based synchronization module for synchronizing the data stored at the remote site with the manipulated data

16 The system of claim 11, wherein the workspace data manager creates another instance of the interface to enable presentation of the downloaded data

17. The system of claim 11, wherein the workspace data manager provides access to its only interface to enable presentation of the downloaded data.

18. The system of claim 11, further comprising a deinstantiator for deleting the interface after it is no longer required.

19. The system of claim 10, further comprising a data reader for translating the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager.

20. A system comprising:

means for executing a workspace data manager on an untrusted client site;

means for requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

means for initiating a communications channel with the remote site;

means for downloading data from the remote site;

means for placing the data in storage on the untrusted client site;

means for using the workspace data manager to present the downloaded data; and

means for disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

21. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : **6,151,606**

DATED : **November 21, 2000**

INVENTOR(S) : **Daniel J. Mendez**

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 12, line 47, after "manipulate the" change the word "dow" to -- downloaded --
Column 13, line 5, after "module for" change the word"download" to -- downloading --
Column 13, line 23, before "manager" insert -- data --
Column 14, line 18, after "placing the data in" insert -- temporary --

Signed and Sealed this

Eighth Day of May, 2001

Attest:

Attesting Officer

NICHOLAS P. GODICI

Acting Director of the United States Patent and Trademark Office

EXHIBIT D

US006023708A

United States Patent [19]

Mendez et al.

[11] Patent Number: 6,023,708

[45] Date of Patent: Feb. 8, 2000

[54] **SYSTEM AND METHOD FOR USING A GLOBAL TRANSLATOR TO SYNCHRONIZE WORKSPACE ELEMENTS ACROSS A NETWORK**

[75] Inventors: **Daniel J. Mendez**, Mountain View; **Mark D. Riggins**, San Jose; **Prasad Wagle**, Santa Clara; **Christine C. Ying**, Foster City, all of Calif.

[73] Assignee: **Visto Corporation**, Mountain View, Calif.

[21] Appl. No.: **08/865,075**

[22] Filed: **May 29, 1997**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/203**; 707/4; 707/201; 707/10; 395/200.35

[58] **Field of Search** 707/10, 203, 201, 707/100, 4; 395/200.36, 200.48, 200.35

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,831,582	5/1989	Miller et al.	707/104
4,897,781	1/1990	Chang	364/200
5,263,157	11/1993	Janis	707/9
5,386,564	1/1995	Shearer et al.	395/650
5,392,390	2/1995	Crozier	395/161
5,572,643	11/1996	Judson	395/793
5,581,749	12/1996	Hossain et al.	707/1
5,613,012	3/1997	Hoffman et al.	382/115
5,623,601	4/1997	Vu	395/187.01
5,627,658	5/1997	Connors et al.	358/407
5,634,053	5/1997	Noble et al.	707/4
5,647,002	7/1997	Brunson	380/49
5,652,884	7/1997	Palevich	395/651
5,666,553	9/1997	Crozier	395/803
5,680,542	10/1997	Mulchandani	395/183.04

(List continued on next page)

OTHER PUBLICATIONS

Margaret J. Brown, "The Visto Briefcase Pro Puts Your PIM On The Internet", URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html, Aug. 13, 1998, 1 page.

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome—Jul. 1997, URL=http://www.bns.it/emware/Bookmark Translator-uk.htm, pp. 1–4.

Article by Bellovin et al., entitled: "Network Firewalls" Published by IEEE Communications Magazine Sep. 1994, pp. 50–57.

Article by Steffen Stempel, entitled: "IPAccess—An Internet Service Access System for Firewall Installations" Published by IEEE Communications Magazine Feb. 16, 1995, pp. 31–41

Article by Braun et al., entitled: "Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server" Published by Elsevier Science B.V. 1995 pp. 37–51

Article by Nelson et al., entitled: "Security for Infinite Networks" Published by IEEE Communications Magazine on Aug. 22, 1995, pp. 11–19

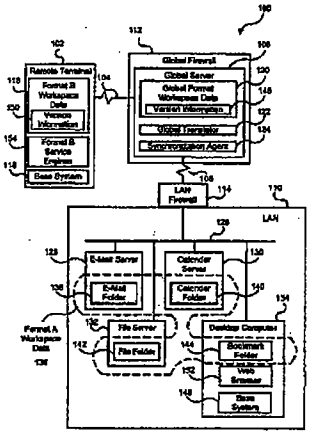
(List continued on next page)

Primary Examiner—Thomas G. Black
Assistant Examiner—Charles L. Rones
Attorney, Agent, or Firm—Graham & James LLP

[57] **ABSTRACT**

A system uses a global translator to automatically synchronize multiple copies of a workspace element across different formats between multiple sites in a secure network environment, independent of whether the sites are protected by site firewalls. The secure network environment includes a global server connected to multiple clients. The system includes a first store for storing a first workspace element in a first format, a second store for storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format, a communications channel coupling the first store to the second store, synchronization means for synchronizing the first workspace element and the second workspace element, and a translator for translating between the first format and the second format.

34 Claims, 7 Drawing Sheets



U.S. PATENT DOCUMENTS

5,682,524	10/1997	Freund et al.	395/605
5,684,990	11/1997	Boothby	707/203
5,701,423	12/1997	Crozier	395/335
5,706,502	1/1998	Foley et al.	395/610
5,713,019	1/1998	Keaten	395/610
5,715,403	2/1998	Stefik	705/44
5,717,925	2/1998	Harper et al	395/613
5,721,914	2/1998	DeVries	395/615
5,729,735	3/1998	Meyering	395/610
5,745,360	4/1998	Leone et al.	364/140
5,752,246	5/1998	Rogers et al.	707/10
5,757,916	5/1998	MacDoran et al	380/25
5,758,150	5/1998	Bell et al.	395/610
5,758,354	5/1998	Huang et al.	707/201
5,758,355	5/1998	Buchanan	707/201
5,765,171	6/1998	Gehani et al	707/203
5,778,346	7/1998	Frid-Nielsen et al.	395/208
5,790,425	8/1998	Wagle	1/1
5,832,483	11/1998	Barker	707/8
5,870,759	2/1999	Bauer et al.	707/201
5,961,590	10/1999	Mendez et al	709/206

OTHER PUBLICATIONS

Article by Greenwald et al., entitled: "Designing an Academic Firewall: Policy, Practice, and Experience with SURF" Published by IEEE Communications Magazine on Feb. 22, 1996, pp. 79–92.

Article by Kiuchi et al., entitled: "C-HTTP—The Development of a Secure, Closed HTTP-based Network on the Internet" Published by IEEE Proceedings of SNDSS on Feb 22, 1996, pp 64–75.

Article by S. Cobb, entitled: "Establishing Firewall Policy" Published by National Computer Security Assn. on Jun. 25–27, 1996, pp. 198–205.

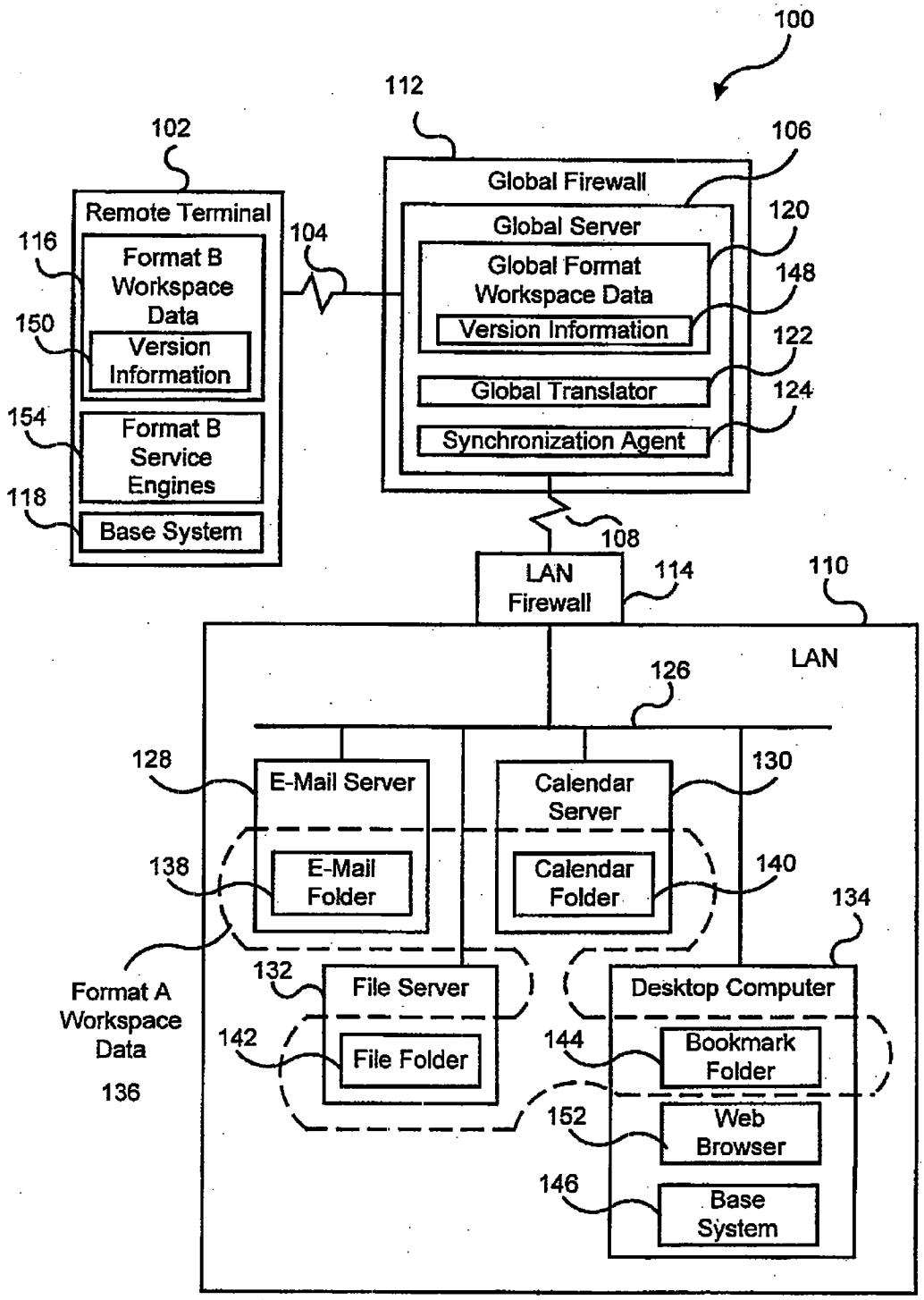

100
112
102
106
Global Firewall
Remote Terminal
120
Global Server
116
Format B Workspace Data
104
Global Format Workspace Data
148
150
Version Information
Version Information
122
Global Translator
154
124
Format B Service Engines
Synchronization Agent
118
Base System
108
LAN Firewall
114
110
LAN
126
128
E-Mail Server
Calendar Server
130
138
E-Mail Folder
Calendar Folder
140
134
Format A Workspace Data
136
132
File Server
Desktop Computer
142
File Folder
144
Bookmark Folder
152
Web Browser
146
Base System

FIG. 1

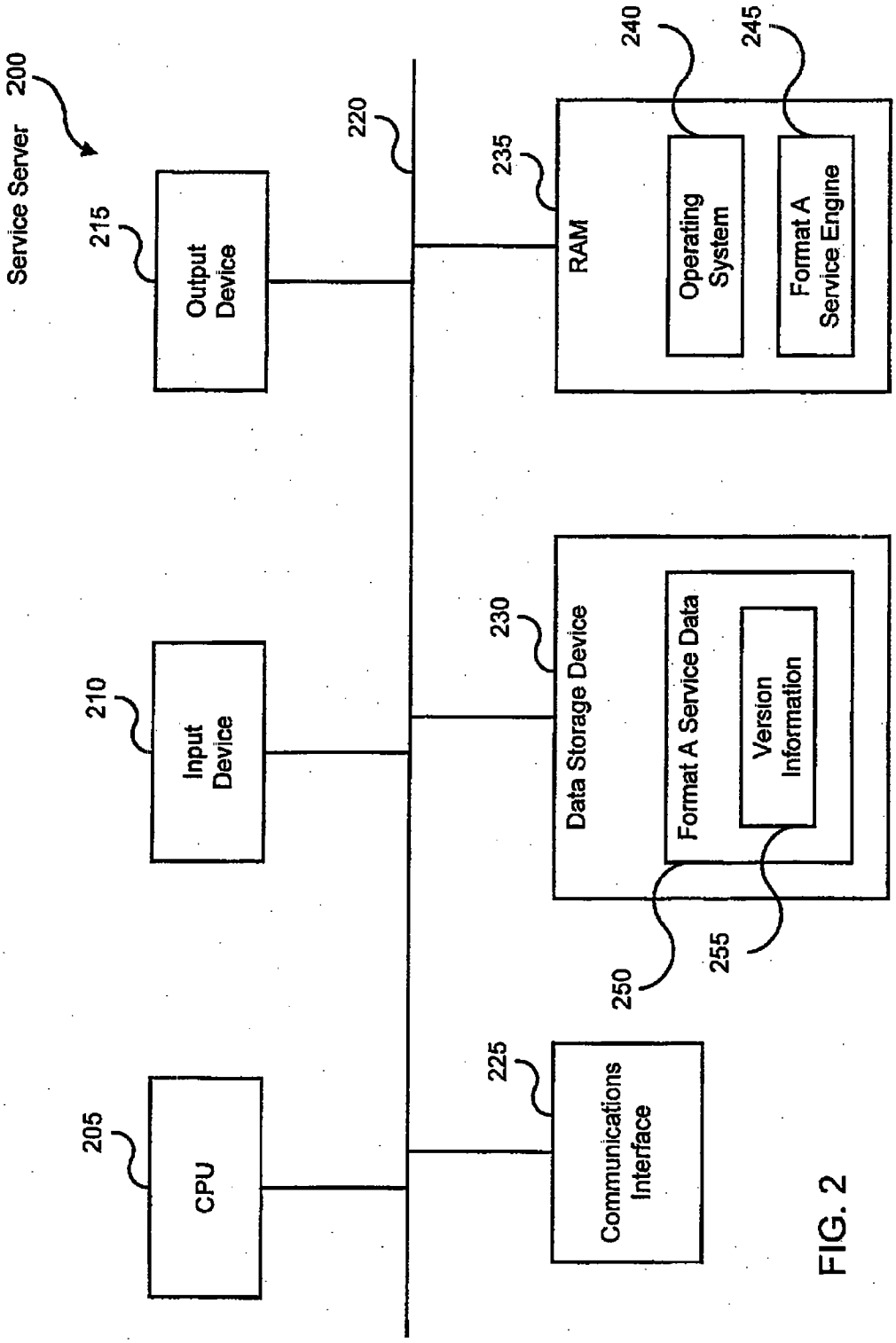

Service Server
200
220
215
Output Device
235
RAM
240
Operating System
245
Format A Service Engine
210
Input Device
230
Data Storage Device
Format A Service Data
Version Information
250
255
205
CPU
225
Communications Interface

FIG. 2

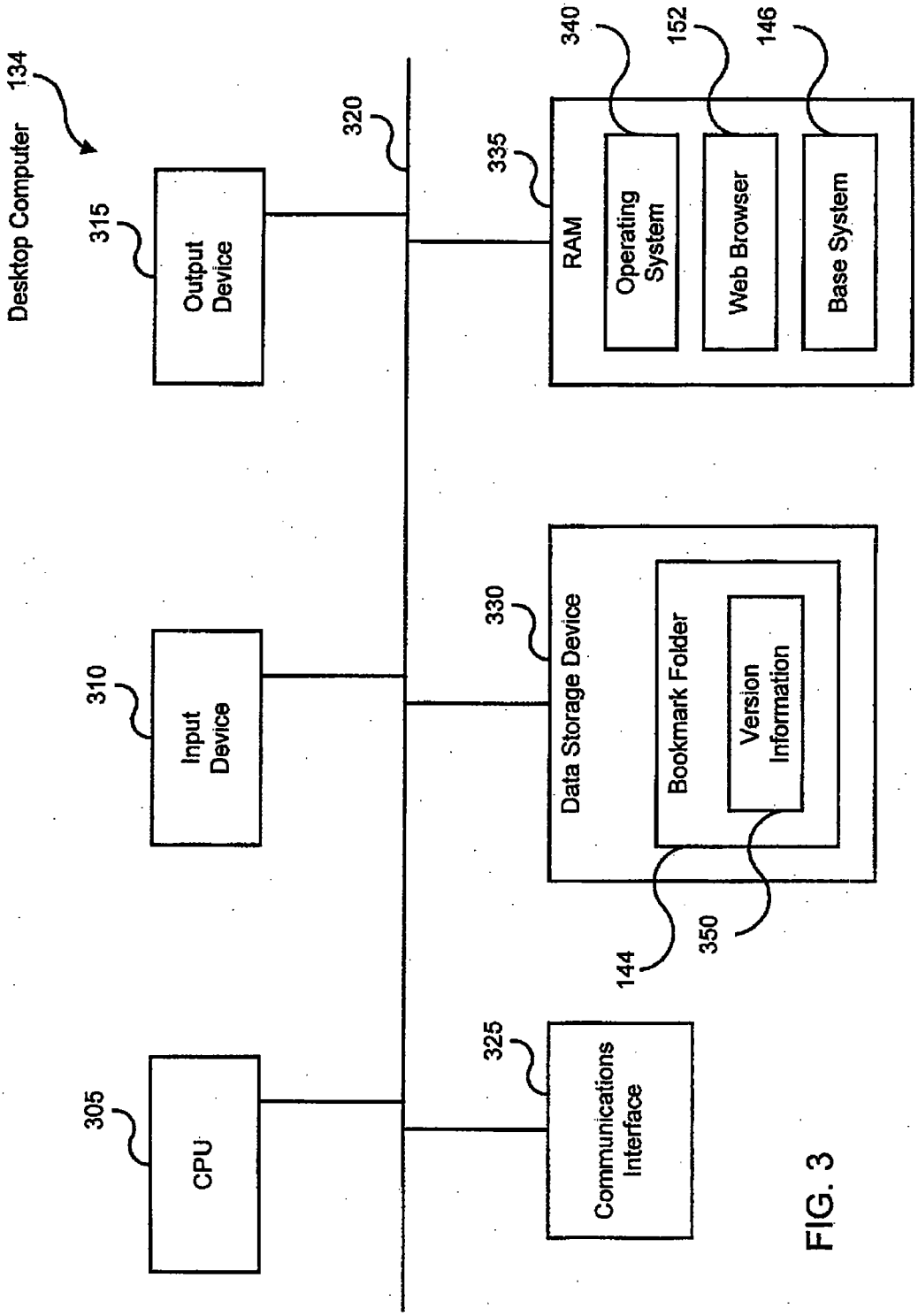

Desktop Computer
134
320
315
Output Device
335
RAM
340
Operating System
152
Web Browser
146
Base System
310
Input Device
330
Data Storage Device
Bookmark Folder
Version Information
144
350
305
CPU
325
Communications Interface

FIG. 3

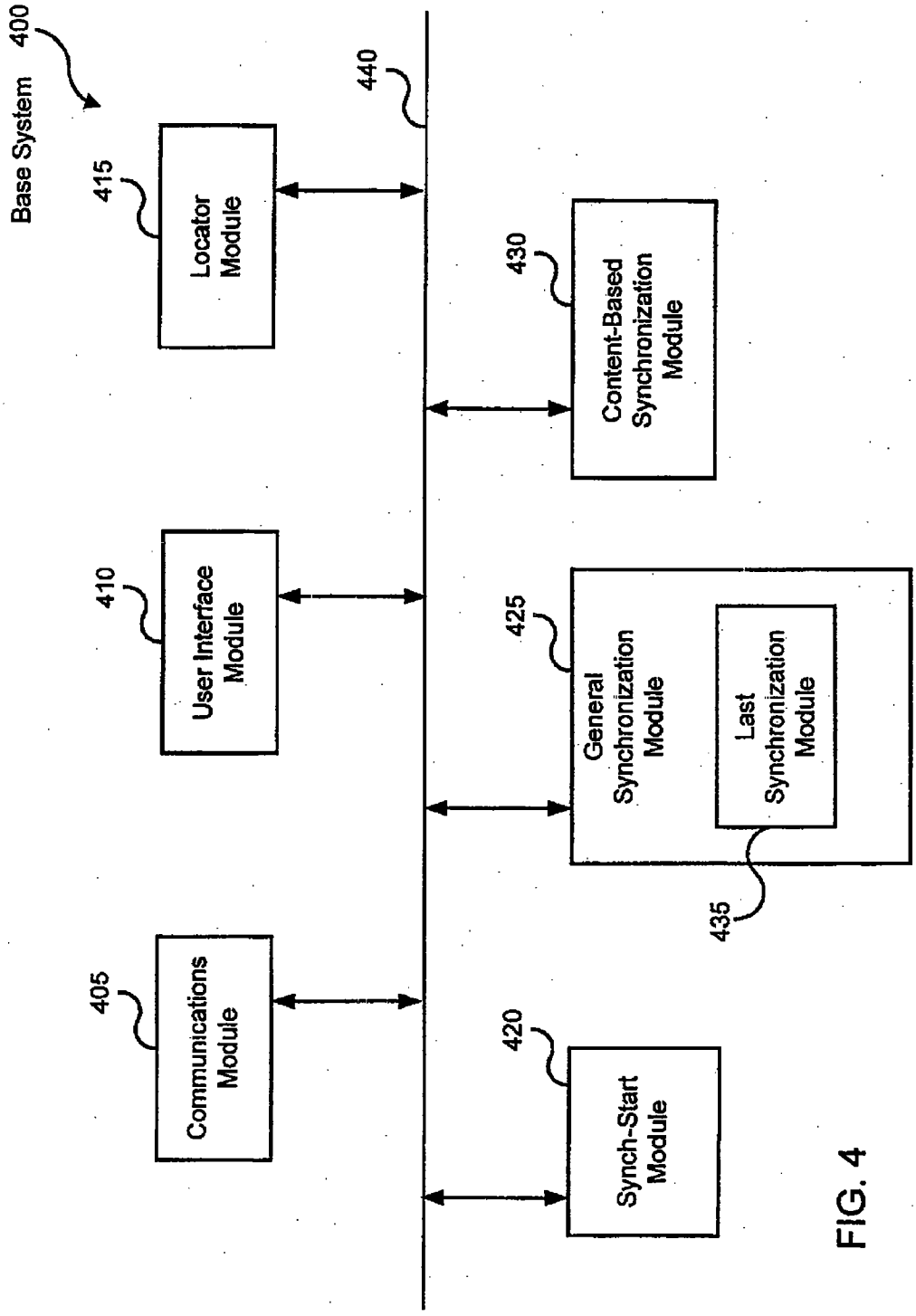

Base System
400
440
415
Locator Module
430
Content-Based Synchronization Module
410
User Interface Module
425
General Synchronization Module
Last Synchronization Module
435
405
Communications Module
420
Synch-Start Module

FIG. 4

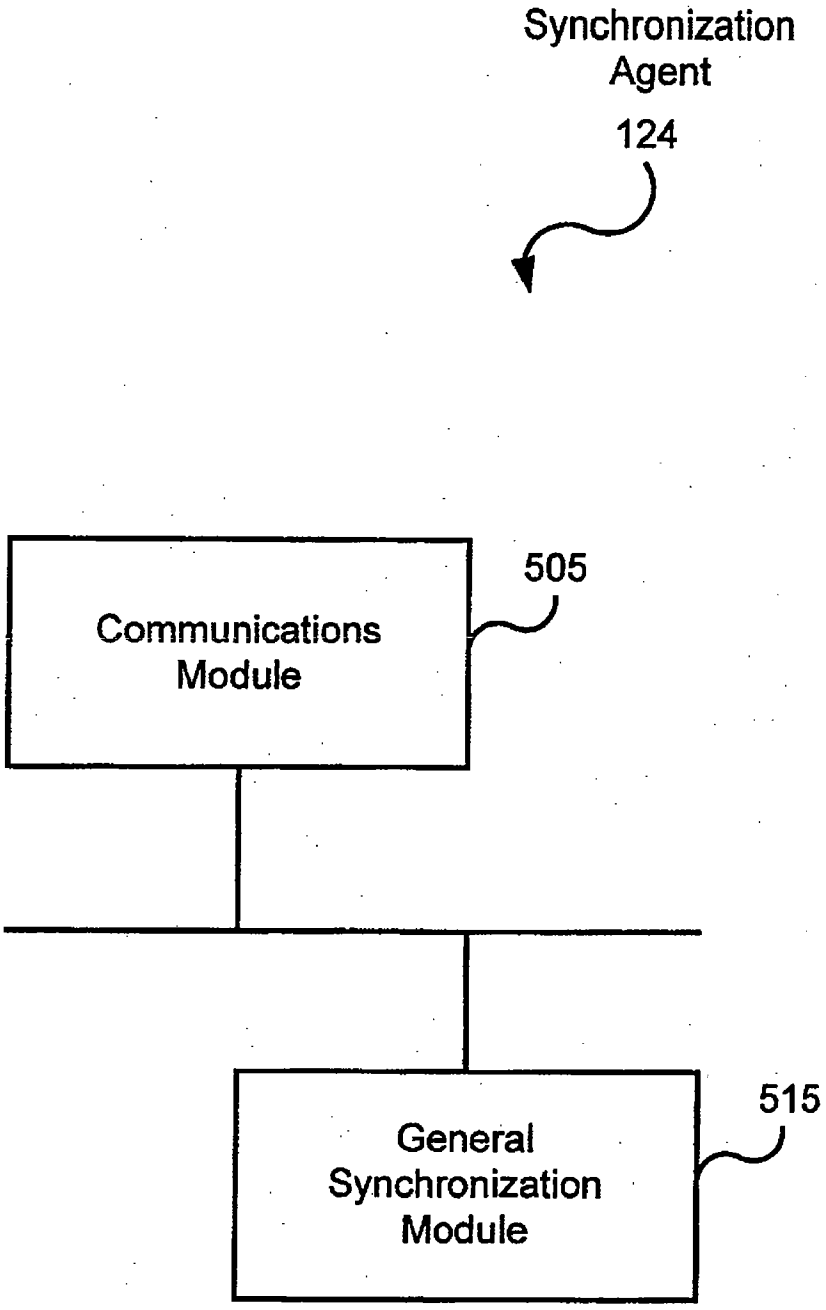

Synchronization Agent
124
Communications Module
505
General Synchronization Module
515

FIG. 5

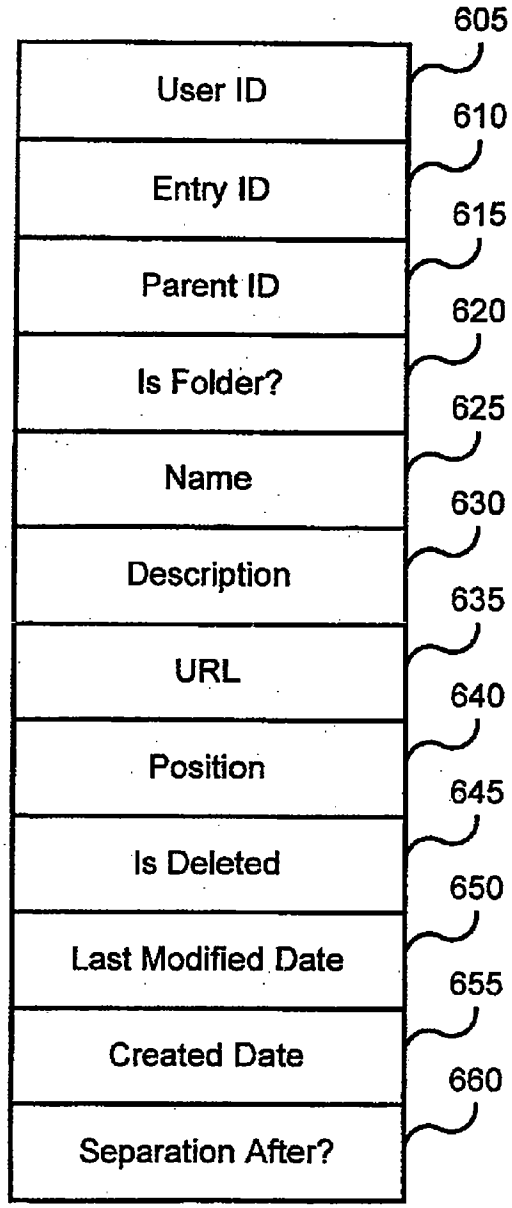


FIG. 6
(Global Format Bookmark Example)

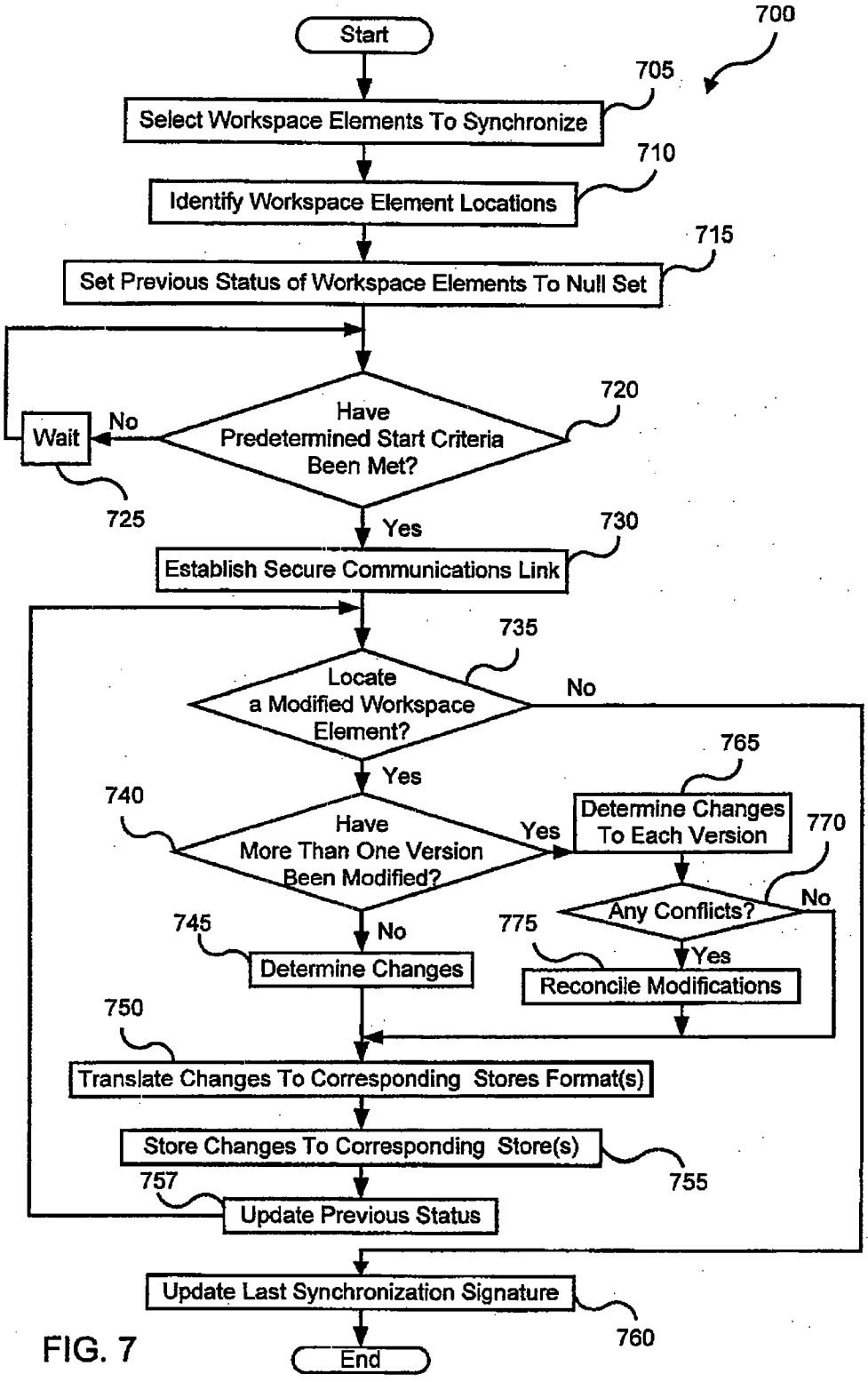

700
Start
705
Select Workspace Elements To Synchronize
710
Identify Workspace Element Locations
715
Set Previous Status of Workspace Elements To Null Set
720
Have Predetermined Start Criteria Been Met?
No
Wait
725
Yes
730
Establish Secure Communications Link
735
Locate a Modified Workspace Element?
No
Yes
740
Have More Than One Version Been Modified?
Yes
765
Determine Changes To Each Version
770
Any Conflicts?
No
Yes
775
Reconcile Modifications
No
745
Determine Changes
750
Translate Changes To Corresponding Stores Format(s)
Store Changes To Corresponding Store(s)
755
757
Update Previous Status
Update Last Synchronization Signature
760
End

FIG. 7

SYSTEM AND METHOD FOR USING A GLOBAL TRANSLATOR TO SYNCHRONIZE WORKSPACE ELEMENTS ACROSS A NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to and hereby incorporates by reference the co-pending patent application entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," Ser. No. 08/835,997, filed on Apr. 11, 1997, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying. This related application has also been assigned to RoamPage, Inc.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for using a global translator to synchronize workspace elements such as files across a computer network

2 Description of the Background Art

Data consistency is a significant concern for computer users. For example, when maintaining multiple independently-modifiable copies of a document, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may have already resulted. The user must then spend more time attempting to reconcile the inconsistent versions and addressing any miscommunications

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

Data consistency problems may also arise when using application programs from different vendors For example, the Netscape Navigator™ web browser and the Internet Explorer™ web browser each store bookmarks for quick reference to interesting web sites. However, since each web browser uses different formats and stores bookmarks in different files, the bookmarks are not interchangeable In addition, one web browser may store a needed bookmark, and the other may not A user who, for example, runs the Internet Explorer™ web browser at home and runs the Netscape Navigator™ web browser at work risks having inconsistent bookmarks at each location

Therefore, a system and method are needed for providing users with data consistency, and more particularly for synchronizing multiple copies of a workspace element such as a document across a computer network

SUMMARY OF THE INVENTION

The present invention provides a system and method for using a global translator to synchronize multiple copies of a workspace element in a secure network environment. The secure network environment includes a global server connected to multiple clients. Using the present system and method, the clients automatically synchronize workspace elements between multiple sites, independent of whether the sites are protected by site firewalls. Using the present system and method, the clients can automatically synchronize workspace elements across different formats and can merge workspace element folders for cross use

The system includes a first store for storing first workspace elements in a first format, a second store for storing second workspace elements in a second format, a communications channel coupling the first store to the second store, synchronization means for synchronizing first workspace elements and second workspace elements, and a translator for translating between the first format and the second format

Similarly, the method includes the steps of accessing a first store storing a first workspace element in a first format, accessing a second store storing a second workspace element in a second format, synchronizing the first workspace element and the second workspace element, and translating between the first format and the second format.

The system and method advantageously use a trusted third party to enable the synchronization of workspace data among multiple sites. Accordingly, a client user who maintains a work site, a home site, an off-site and the global server site can synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which control when the synchronization-start module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, synchronization can occur over any distance. Since the system and method include format translation, merging of workspace elements between different application programs and different platforms is possible Further, because synchronization is initiated from within the firewall, the typical firewall which prevents inbound communications does not act as an impediment to workspace element synchronization. Also, since the user's preferences may be previously set, the present system and method may operate unattended by the client user.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a computer network in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of a FIG. 1 service server;

FIG. 3 is a block diagram illustrating details of the FIG. 1 desktop computer;

FIG. 4 is a block diagram illustrating details of a FIG. 1 base system;

FIG. 5 is a block diagram illustrating details of the FIG. 1 synchronization agent;

FIG. 6 is a graphical representation of an example bookmark in the global format; and

FIG. 7 is a flowchart illustrating a method for synchronizing multiple copies of a workspace element in a secure network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a computer network **100**, comprising a first node such as a remote computer terminal **102** coupled via a communications channel **104** such as the Internet to a global server **106**. The global server **106** is in turn coupled via a communications channel **108** such as the Internet to a second node such as a Local Area

Network (LAN) **110**. The global server **106** is protected by a global firewall **112**, and the LAN **110** is protected by a LAN firewall **114**.

The LAN **110** includes a system bus **126** coupling the LAN firewall **114** to an e-mail server **128** having an e-mail folder **138** containing e-mails, to a file server **132** having a file folder **142** containing files, to a calendar server **130** having a calendar folder **140** containing calendar data, and to a desktop computer **134** having a web browser **152** and a bookmark folder **144** containing bookmarks. It will be appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** or portions thereof may be stored at different locations such as on the desktop computer **134**. The e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** are exemplary, grouped by like information and are collectively referred to herein as "workspace data" **136**. Those skilled in the art will recognize that the workspace data **136** may include other types of data such as an application program such as Microsoft Word 6.0 1 TM word processor and the documents created using them. It will be further appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** may each be divided into workspace elements, wherein each workspace element folder or each workspace element individually is identified by particular version information **255** (described below with reference to FIG 2). Accordingly, each e-mail or e-mail folder, file or file folder, calendar or calendar folder, bookmark or bookmark folder, document or document folder, etc may be referred to as "a workspace element."

Each workspace element of workspace data **136** in LAN **110** is maintained in a predetermined format, referred to as Format A, which is based on the service engine **245** (FIG 2) that created it For example, the web browser **152** on the desktop computer **134** may be the Netscape Navigator™ web browser, and the bookmarks in the bookmark folder **144** created thereby are maintained in Format A. Although Format A is being described as a single format, one skilled in the art knows that Format A actually includes a format for each information type, e.g., there will be a Format A for bookmarks, a Format A for files, a Format A for calendar data, a Format A for e-mails, etc

The remote terminal **102** stores service engines **154** for maintaining workspace data **116**, which may include information common with information in the workspace data **136**. The workspace data **116** is maintained in a format, referred to as Format B, which may be different from Format A Format B is also based on the service engines **154** that create the workspace elements. For example, if one of the service engines **154** is the Internet Explorer™ web browser (not shown), then the bookmarks (not shown) created therewith are maintained in Format B. Although Format B is being described as a single format, one skilled in the art knows that Format B actually includes a format for each information type. Further, the workspace data **116** also includes version information **150** similar to version information **255** described below with reference to FIG. 2.

It will be appreciated that remote terminal **102** may include a smart telephone, a Personal Data Assistant (PDA) such as the PalmPilot TMPDA the U.S. Robotics, Inc., a laptop computer, etc. As a smart telephone, the workspace data **116** may include telephone numbers and e-mails. As a PDA, the workspace data **116** may include addresses, calendar data and e-mails. As a laptop computer, the workspace data **116** may include the same types of information as workspace data **136**.

The global server **106** acts as a third party administrator The global server **106** stores independently-modifiable copies of selected portions of the workspace data **136** and **116**, collectively referred to herein as workspace data **120**. Accordingly, the workspace data **120** includes an independently-modifiable copy of each workspace element in the selected portions of the workspace data **136** and **116** and an independently-modifiable copy of each corresponding version information **255** (FIG. 2) and **150**. The version information copies are collectively referred to herein as version information **148**, and are also described with reference to FIG 2.

The global server **106** maintains the workspace data **120** in a format, referred to as a "global format," which is selected to be easily translatable by the global translator **122** to and from Format A and to and from Format B. Although the global format is being described as a single format, one skilled in the art knows that the global format actually includes a global format for each information type, e.g., there will be a global format for bookmarks, a global format for files, a global format for calendar data, a global format for e-mails, etc. An example bookmark workspace element in the global format is described in detail below with reference to FIG. 6.

Network **100** further comprises synchronization means, which includes a base system **146** stored within the LAN **110** and for example on the desktop computer **134**. Network **100** further includes a synchronization agent **124** stored outside the LAN firewall **114** and preferably on the global server **106** The base system **146** and the synchronization agent **124** cooperate to synchronize selected portions of the workspace data **136** with selected portions of the workspace data **120** The synchronization means may synchronize workspace elements individually, e.g., specific word processor documents, or may synchronize workspace element folders, e.g., a bookmark folder. Generally, the base system **146** manages the selected portion of the workspace data **136** within the LAN **110** and the synchronization agent **124** manages the selected portions of the workspace data **120** within the global server **106**. It will be appreciated that the global translator **122** cooperates with the synchronization means to translate data formats to and from the global format. As described in greater detail below with reference to FIG 4, the base system **190** preferably initiates and controls data synchronization. Other components and functions of the global server **106** are described in the cross-referenced patent application which is herein incorporated by reference.

The synchronization means may also include, stored on the remote terminal **102**, a base system **118** which operates in a similar manner to the base system **146**. The base system **118** on the remote terminal **102** cooperates with the synchronization agent **124** to synchronize selected portions of the workspace data **116** with selected portions of the workspace data **120**. As described in greater detail below with reference to FIG. 4, the base system **118** on the remote terminal **102** also preferably initiates and controls data synchronization with the global server **106**. Also, note that the distribution of labor between the base system **118** in the remote terminal **102** and the synchronization agent **124** in the global server **106** may vary. Sometimes, primarily when the remote terminal **102** is a relatively less computationally powerful device (such as a smart phone or a PDA), most of the actual computationally intensive work will occur within the synchronization agent **124** in the global server **106**. In other situations, for example, when the remote terminal **102** is a fully configured PC, most of the computationally intensive work will occur locally on the base system **118** in the remote terminal **102**.

Accordingly, the synchronization means independently synchronizes the selected portions of workspace data **116** and **136** with the selected portions of the workspace data **120**. Thus, the synchronization means indirectly synchronizes workspace data **136** with workspace data **116**.

FIG. **2** is a block diagram illustrating details of a service server **200**, wherein each of the e-mail server **145**, the file server **150**, the calendar server **155** and the desktop computer **160** is an instance thereof. Service server **200** includes a Central Processing Unit (CPU) **205** such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor. An input device **210** such as a keyboard and mouse and an output device **215** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **220** to CPU **205**. A communications interface **225** (such as an Ethernet port), a data storage device **230** (such as a magnetic disk), and Random-Access Memory (RAM) **235** are further coupled via signal bus **220** to the CPU **205**.

An operating system **240** includes a program for controlling processing by the CPU **205**, and is typically stored in the data storage device **230** and loaded into the RAM **235** for execution. A service engine **245** includes a program for performing a particular service such as maintaining an e-mail data base, a file data base, a calendar data base or a bookmarks data base. The service engine **245** may also be stored in the data storage device **230** and loaded into the RAM **235** for execution.

To perform a service, the service engine **245** creates service data **250** (e.g., an e-mail or an e-mail folder **138** containing e-mails, a file or a file folder **142** containing files, calendar data or a calendar folder **140** containing calendar data, a bookmark or a bookmark folder **144** containing bookmarks, etc.) in Format A according to predetermined protocols. The service engine **245** stores the data **250** in the data storage device **250**. The service data **250** includes version information **255** indicating the date and time of the last modification and the status as of the last interaction with the global server **106**.

For example, if service data **250** is created and selected to be merged with global server workspace data **120**, then the version information **255** for the service data **250** may include the date of last modification and a null set indicating the status as of the last interaction with the global server **106**. From the version information **255**, the base system **146** determines that the service data **250** in its entirety has not been merged with the global server workspace data **120**. Similarly, if the service data **255** included elements **1, 2** and **3** as of the last modification, then the previous status as of the last interaction will indicate that the service data **255** included elements **1, 2** and **3**. If the service data **255** currently includes elements **2, 3** and **4**, then the base system **140** will determine that, since last synchronization, element I has been deleted and element **4** has been added.

It will be appreciated that the version information **148** on the global server **106** includes information similar to version information **255**. That is, the version information **148** will include information indicating the date and time the version was last modified and the status as of the last interaction with each client. The service engine **245** operates to update the version information **255** after modifications are made and after synchronization occurs.

FIG. **3** is a block diagram illustrating details of the desktop computer **160**, which includes a CPU **305**, an input device **310**, an output device **315**, a communications interface **325**, a data storage device **330** and RAM **335**, each coupled to a signal bus **320**.

An operating system **340** includes a program for controlling processing by the CPU **305**, and is typically stored in the data storage device **330** and loaded into the RAM **335** for execution. A web browser **152** (i.e., a particular service engine **245**, FIG. **2**) includes a Format A service program for managing bookmark folder **144** (i.e., particular service data **250**, FIG. **2**) which includes, version information **350** (i.e., particular version information **255**, FIG. **2**). The web browser **152** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution. The bookmark folder **144** may be stored in the data storage device **330**. As stated above with reference to FIG. **1**, the base system **146** operates to synchronize the workspace data **136** (which includes the bookmark folder **144**) with the workspace data **120**. The base system **146** may be also stored in the data storage device **330** and loaded into the **5** RAM **335** for execution.

FIG. **4** is a block diagram illustrating details of the base system **400**, which exemplifies base systems **146** and **118**. Base system **400** includes a communications module **405**, a user interface module **410**, locator modules **415**, a synchronization-start ("synch-start") module **420**, a general synchronization module **425** and a content-based synchronization module **430**. For simplicity, each module is illustrated as communicating with one another via a signal bus **440**.

The communications module **405** includes routines for compressing data and routines for communicating via the communications interface **325** (FIG. **3**) with the synchronization agent **124** (FIG. **1**). The communications module **405** may further include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the global firewall **112**. Examples of communications modules **405** may include TCP/IP stacks or the AppleTalk™ protocol.

The user interface module **410** includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface module **410** cooperates with the other system components as described herein.

The locator modules **415** include routines for identifying the memory locations of the workspace elements in the workspace data **136** or **116** and in the workspace data **120**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query a user. More particularly, the locator modules **415** in the base system **146** determine the memory addresses of the e-mail folder **138**, the file folder **142**, the calendar folder **140** and the bookmark folder **144** and the memory addresses of the workspace elements therein. The locator modules **415** also determine the corresponding memory addresses of the corresponding folders in the workspace data **120** and the corresponding workspace elements therein. Similarly, the locator modules **415** in the base system **118** determine the memory locations of the workspace elements of workspace data **116** and the memory locations of the corresponding workspace elements in the workspace data **120**.

It will be appreciated that the locator modules **415** may include locator modules **415** specifically dedicated to each folder or workspace data type. That is, the locator modules **415** may include a locator module **415** dedicated to locating bookmarks, a locator module **415** dedicated to locating e-mails, a locator module **415** dedicated to locating files, a locator module **415** dedicated to locating calendar

appointments, etc. It will be further appreciated that the locator modules **415** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **120** to maintain updated memory addresses of workspace elements.

The synchronization-start module **420** includes routines for determining when to initiate synchronization of workspace data **136** or **116** with workspace data **120**. For example, the synchronization-start module **420** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria. The synchronization-start module **420** initiates data synchronization by instructing the general synchronization module **425** (described below) to begin execution of its routines. It will be appreciated that communication with the synchronization agent **124** preferably initiates from within the LAN **10**, because the typical firewall **114** prevents in-bound communications and allows out-bound communications.

The general synchronization module **425** includes routines for receiving version information **148** for modified versions from the synchronization agent **124** (FIG. 1), and routines for examining the version information **255** or **150** against a last synchronization signature **435** (such as a last synchronization date and time) to determine which versions have been modified. The general synchronization module **425** further includes routines for examining the version information **148** and the version information **255** or **150** to determine if one or both versions of a particular workspace element or workspace element folder have been modified.

Further, the general synchronization module **425** includes routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include, if only one version of a workspace element in workspace data **136** or **116** has been modified, then forwarding the modified version (as the preferred version) to the other store(s) or determining and forwarding only the changes made. Computing the changes made may be performed by examining the current status against the previous status as of the last synchronization or by comparing the two versions. It will be appreciated that no content-based review of the changes is needed. It will be appreciated that one store preferably forwards only the changes to the other store for optimizing use of processor power and minimizing the data communications across the communications channel **108** or **104**.

Other appropriate synchronizing responsive actions may include, if two versions of a workspace element have been modified independently, then instructing the content-based synchronization module **430** (described below) to execute its routines. That is, if two versions of the same workspace element have been modified independently, then a content-based review of the changes is preferable. Upon completion of the data synchronization, the general synchronization module **425** updates the last synchronization signature **435**.

The content-based synchronization module **430** includes routines for reconciling two or more modified versions of a workspace element. For example, if a user has independently modified the original and the copy of a workspace element since the last synchronization, then the content-based synchronization module **430** determines an appropriate responsive action. The content-based synchronization module **430** may request the user to select a preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or preferably by integrating the modified versions into a single preferred version which replaces each modified version at both stores.

The content-based synchronization module **430** examines the changes made to each version and determines if conflicts exist. When implementing version integration, a conflict may arise if inconsistent modifications such as deleting a paragraph in one version and modifying the same paragraph in the other version have been made. If a conflict exists, then the content-based synchronization module **430** attempts to reconcile the conflict, e.g., by requesting user selection or by storing both versions at both stores. Otherwise, if no conflict exists, then the content-based synchronization module **430** integrates the changes to each of the versions and updates the version information **148**, **150** or **255** accordingly.

FIG. **5** is a block diagram illustrating details of the synchronization agent **124**, which includes a communications module **505** (similar to the communications module **405** described above with reference to FIG. **4**) and a general synchronization module **515** (similar to the general synchronization module **425** described above also with reference to FIG. **4**).

The communications module **505** includes routines for compressing data, and routines for communicating via the communications channel **108** with the base system **146** or via the communications channel **104** with the base system **118**. The communications module **505** may further include routines for establishing a secure communications channel through the global firewall **112** and through the LAN firewall **114** with the communications module **405**.

Similar to the general synchronization module **425**, the general synchronization module **515** includes routines for examining the version information **148** and the last synchronization signature **435** (FIG. 4) to determine which versions have been modified and the changes made. It will be appreciated that the general synchronization module **515** may maintain its own last synchronization signature **435** copy (not shown) or may request the last synchronization signature **435** from the base system **146** or **118**. The general synchronization module **515** further includes routines for forwarding workspace data **120** determined to be modified to the general synchronization module **425**, and routines for receiving preferred versions of workspace elements of workspace data **136** or **116** or just the changes from the general synchronization module **425**.

FIG. **6** illustrates an example bookmark workspace element in the global format. The global translator **122** incorporates all the information needed by both formats (Format A and Format B) to create the Global Format. For example, if a bookmark in Format A needs elements X, Y and Z and a bookmark in Format B needs elements W, X and Y, the global translator **122** incorporates elements W, X, Y and Z to create a bookmark in the Global Format. Further, the global translator **122** incorporates the information which is needed by the synchronization means such as the last modified date. Accordingly, a bookmark in the Global Format includes a user identification (ID) **605**, an entry ID **610**, a parent ID **615**, a folder ID flag **620**, a name **625**, a description **630**, the Uniform Resource Locator (URL) **635**, the position **640**, a deleted ID flag **645**, a last modified date **650**, a created date **655** and a separation ID flag **660**.

FIG. **7** is a flowchart illustrating a method **700** for using a global translator **122** to synchronize multiple copies of a workspace element in a secure network **100**. Method **700** begins with the user interface module **410** in step **705** enabling a user to select workspace elements of workspace

data 136 and 118 for the synchronization means to synchronize The locator modules 415 in step 710 identify the memory locations of the workspace elements in workspace data 136 and 116 and the corresponding memory locations in workspace data 120. If a selected workspace element does not have a corresponding memory location, such as in the case of adding a new workspace elements to the global server 106, then one is selected. The selected memory location may be a preexisting workspace element or a new workspace element. As stated above, workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The general synchronization module 425 and general synchronization module 515 in step 715 set the previous status of the workspace elements equal to the null set Setting the previous status to the null set indicates that all information of the workspace element has been added.

The synchronization-start module 420 in step 720 determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step 705 should start. If not, then the synchronization-start module 420 in step 725 waits and loops back to step 720. Otherwise, the communications module 405 and communications module 505 in step 730 establish a secure communications channel therebetween

The general synchronization module 425 and the general synchronization module 515 in step 735 determine whether any workspace elements have been modified. That is, the general synchronization module 425 in step 740 examines the version information 255 or 150 of each selected workspace element in the workspace data 136 or 116 against the last synchronization signature 435 to locate modified workspace elements. This comparison may include comparing the date of last modification with the date of last synchronization, or may include a comparison between the current status and the previous status as of the last interaction. Similarly, the general synchronization module 515 examines the version information 148 of each corresponding workspace element in workspace data 120 and the last synchronization signature 435 to locate modified workspace elements.

If in step 735 no modified workspace elements or folders are located, then the general synchronization modules 425 and 515 in step 760 update the last synchronization signature. 435 and method 700 ends. Otherwise, the general synchronization module 425 in step 740 determines whether more than one version of a workspace element has been modified since the last synchronization.

If only one version has been modified, then the corresponding general synchronization module 425 or 515 in step 745 determines the changes made. As stated above, determining the changes made may be implemented by comparing the current status of the workspace element against the previous status of the workspace element as of the last interaction therebetween. If the changes were made only to the version in the workspace data 120, then the global translator 122 in step 750 translates the changes to the format used by the other store, and the general synchronization module 515 in step 755 forwards the translated changes to the general synchronization module 425 for updating the outdated workspace element in the workspace data 136 or 116. If the updated version is a workspace element in the workspace data 136 or 116, then the general synchronization module 425 sends the changes to the updated version to the global translator 122 for translation and then to the general synchronization module 515 for updating the outdated workspace element in the workspace data 120. The general synchronization module 425 and the general synchronization module 515 in step 757 update the previous state of to reflect the current state as of this interaction Method 700 then returns to step 735.

If the general synchronization module 425 in step 740 determines that multiple versions have been modified, then the general synchronization module 425 in step 765 computes the changes to each version and in step 770 instructs the content-based synchronization module 430 to examine content to determine if any conflicts exist. For example, the content-based synchronization module 430 may determine that a conflict exists if a user deletes a paragraph in one version and modifies the same paragraph in another version The content-based synchronization module 430 may determine that a conflict does not exist if a user deletes different paragraphs in each version. If no conflict is found, then method 700 jumps to step 750 for translating and forwarding the changes in each version to the other store. However, if a conflict is found, then the content-based synchronization module 430 in step 775 reconciles the modified versions. As stated above, reconciliation may include requesting instructions from the user or based on preselected preferences performing responsive actions such as storing both versions at both stores. Method 700 then proceeds to step 750.

It will be appreciated that in step 7 10 new workspace elements and preexisting workspace elements to which new workspace elements will be merged are set to "modified" and the previous status is set to the null set. Thus, the general synchronization module 425 in step 740 will determine that more that one version has been modified and the content-based synchronization module 430 in step 770 will determine that no conflict exists. The changes in each will be translated and forwarded to the other store Accordingly, the two versions will be effectively merged and stored at each store.

For example, if a first bookmark folder was created by the web browser 152 on the desktop computer 134, a second folder was created by a web browser (not shown) on the remote terminal 102, no preexisting folder existed on the global server 106 and the user selected each of these folders for synchronization, then the synchronization means will effectively merge the first and second folders. That is, the general synchronization module 425 on the desktop computer 134 will determine that the first folder has been modified and the previous status is equal to the null set. The general synchronization module 425 will determine and send the changes, i.e., all the workspace elements in the first folder, to a new global folder on the global server 106. Similarly, the general synchronization module 425 on the remote terminal 102 will determine that, as of its last interaction, the previous status of each of the second and the global folders is the null set. The general synchronization module 425 will instruct the content-based synchronization module 430 to examine the changes made to each folder to determine whether a conflict exists. Since no conflicts will exist, the general synchronization module 425 will forward the changes to the global folder and the general synchronization module 515 will forward its changes to the second store, thereby merging the workspace elements of the first and second folders in the global and second folders The general synchronization module 515 will inform the general synchronization module 425 that the global folder has been modified relative to the last interaction, and will forward the new changes to the first folder Thus, the first and second folders will be merged and stored at each store.

For a second example, the user may select an exemplary document in the LAN 110 to be synchronized. The general

synchronization module **425** will forward the document to the global server **106**. Similarly, the user may select the same document for synchronization on the remote terminal **102**. The general synchronization module **515** will forward the document to the remote terminal **102**. If changes were made to the documents independently, then the content-based synchronization module **430** will examine the content of the documents to determine if a conflict exists. If no conflict exists, then as described above, the general synchronization modules **425** and **515** will merge the documents. Otherwise, if a conflict does exist, the content-based synchronization module **430** will reconcile the changes and then the general synchronization modules **425** and **515** will forward the reconciled changes to each other.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, although the global server **106** is illustrated as a single device, the global server **106** may include several computers networked together. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A system, comprising:

a first store for storing a first workspace element in a first format;

a second store for storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

a communications channel coupling the first store to the second store;

synchronization means for synchronizing the first workspace element and the second workspace element; and

a translator for translating between the first format and the second format.

2. The system of claim **1**, wherein the first format is the same as the second format.

3. The system of claim **1** wherein the second format includes HTML.

4. The system of claim **1**, wherein the first store is in a LAN.

5. The system of claim **1**, further comprising a firewall for protecting the first store.

6. The system of claim **1**, further comprising a firewall for protecting the second store.

7. The system of claim **1**, wherein the synchronization means includes

a first general synchronization module for examining the first workspace element at the first store to determine whether it has been modified; and

a second general synchronization module for examining the second workspace element at the second store to determine whether it has been modified.

8. The system of claim **7**, wherein

the first workspace element and the second workspace element each include version information, and

the first and second general synchronization modules each examine the version information of each workspace element to a last synchronization signature to determine whether the workspace element was modified.

9. The system of claim **1**, wherein the synchronization means includes a synchronization-start module for determining when to initiate synchronization.

10. The system of claim **1**, wherein the first workspace element includes a first folder and the second workspace element includes a second folder.

11. The system of claim **10**, further comprising a content-based synchronization module for examining content of the first workspace element and of the second workspace element when the first workspace element and the second workspace element have both been modified since the last synchronization.

12. The system of claim **11**, wherein the content-based synchronization module determines if any conflicts exist, and if so then the content-based synchronization module reconciles the conflicts.

13. The system of claim **11**, wherein the content-based synchronization module determines if any conflicts exist, and if not then the content-based synchronization module integrates the changes made to each workspace element.

14. The system of claim **1**, further comprising a third store coupled to the second store for storing a third workspace element in a third format.

15. The system of claim **14**, wherein the translator translates between the second format and the third format.

16. The system of claim **8**, wherein the version information indicates the status of each workspace element as of the last synchronization.

17. A method, comprising:

accessing a first store storing a first workspace element in a first format;

accessing a second store storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

synchronizing the first workspace element and the second workspace element; and

translating between the first format and the second format.

18. The method of claim **17**, wherein the first format is the same as the second format.

19. The method of claim **17**, wherein the second format includes HTML.

20. The method of claim **17**, wherein the first store is in a LAN.

21. The method of claim **17**, wherein the first store is protected by a firewall.

22. The method of claim **17**, wherein the second store is protected by a firewall.

23. The method of claim **17**, further comprising

examining, the first workspace element to determine whether it has been modified; and

examining the second workspace element to determine whether it has been modified.

24. The method of claim **23**,

wherein the first workspace element and the second workspace element each include version information; and

further comprising the step of comparing the version information of each workspace element to a last synchronization signature to determine whether the workspace element was modified.

25. The method of claim **17**, further comprising the step of determining when to initiate synchronization.

26. The method of claim **17**, wherein the first workspace element includes a first folder and the second workspace element includes a second folder.

27. The method of claim 26, further comprising the step of examining, content of the first workspace element and of the second workspace element when the first workspace element and the second workspace element have both been modified since the last synchronization.

28 The method of claim 27, further comprising the step of determining if any conflicts exist, and if so then reconciling the conflicts

29. The method of claim 27, further comprising the step of determining if any conflicts exist, and if not then integrating the changes made to each workspace element

30. The method of claim 17, further comprising the step of accessing a third store coupled to the second store storing, a third workspace element in a third format.

31. The method of claim 30, wherein further comprising the step of translating between the second format and the third format.

32. The method of claim 24, wherein the version information indicates the status of each workspace element as of the last synchronization.

33. A system, comprising:

means for accessing a first store storing a first workspace element in a first format;

means for accessing a second store storing a second workspace element which is an independently-modifiable copy of the first workspace element in a second format;

means for synchronizing the first workspace element and the second workspace element; and

means for translating between the first format and the second format.

34 A computer-readable storage medium storing program code for causing a computer to perform the steps of:

accessing a first store storing a first workspace element in a first format;

accessing a second store storing a second workspace element which is an independently modifiable copy of the first workspace element in a second format;

synchronizing the first workspace element and the second workspace element; and

translating between the first format and the second format

* * * * *

EXHIBIT E

US006708221B1

(12) **United States Patent**
Mendez et al.

(10) **Patent No.: US 6,708,221 B1**
(45) **Date of Patent: Mar. 16, 2004**

(54) **SYSTEM AND METHOD FOR GLOBALLY AND SECURELY ACCESSING UNIFIED INFORMATION IN A COMPUTER NETWORK**

(75) Inventors: **Daniel J. Mendez**, Menlo Park, CA (US); **Mark D. Riggins**, Mercer Island, WA (US); **Prasad Wagle**, Santa Clara, CA (US); **Hong Q. Bui**, Cupertino, CA (US); **Mason Ng**, Mountain View, CA (US); **Sean Michael Quinlan**, San Francisco, CA (US); **Christine C. Ying**, Foster City, CA (US); **Christopher R. Zuleeg**, San Jose, CA (US); **David J. Cowan**, Menlo Park, CA (US); **Joanna A. Aptekar-Strober**, Menlo Park, CA (US); **R. Stanley Bailes**, San Jose, CA (US)

(73) Assignee: **Visto Corporation**, Redwood Shores, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C 154(b) by 0 days

(21) Appl No : **09/666,877**

(22) Filed: **Sep. 20, 2000**

Related U.S. Application Data

(63) Continuation of application No. 08/903,118, filed on Jul. 30, 1997, and a continuation-in-part of application No. 08/865,075, filed on May 29, 1997, now Pat. No. 6,023,708, and a continuation-in-part of application No. 08/835,997, filed on Apr. 11, 1997, now Pat. No. 6,085,192, and a continuation-in-part of application No. 08/841,950, filed on Apr. 8, 1997, which is a continuation-in-part of application No. 08/766,307, filed on Dec 13, 1996, now Pat. No. 6,131,116.

(51) **Int. Cl.**[7] **G06F 15/16**
(52) **U.S. Cl.** **709/248**; 709/204
(58) **Field of Search** 709/203, 219, 709/248, 245, 204, 205

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,388,255 A * 2/1995 Pytlik et al. 707/4
5,544,320 A * 8/1996 Konrad 709/203

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

CA	2210763	1/1999
EP	0801478	10/1997
WO	WO 99/05620	2/1999
WO	WO 99/45451	9/1999

OTHER PUBLICATIONS

US 5,373,559, 12/1994, Kaufman et al. (withdrawn)

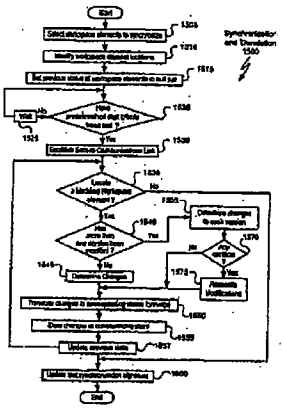
(List continued on next page.)

Primary Examiner—Mehmet B. Geckil
(74) *Attorney, Agent, or Firm*—Squire, Sanders & Dempsey L.L.P.

(57) **ABSTRACT**

A client stores a first set of workspace data, and is coupled via a computer network to a global server. The client may be configured to synchronize portions of the first set of workspace data with the global server, which stores independently modifiable copies of the portions. The global server may also store workspace data which is not downloaded from the client, and thus stores a second set of workspace data. The global server may be configured to identify and authenticate a user seeking global server access from a remote terminal, and is configured to provide access to the first set or to the second set. Further, services may be stored anywhere in the computer network. The global server may be configured to provide the user with access to the services. The system may further include a synchronization-start module at the client site (which may be protected by a firewall) that initiates interconnection and synchronization with the global server when predetermined criteria have been satisfied.

14 Claims, 15 Drawing Sheets



U.S. PATENT DOCUMENTS

5,588,132	A	*	12/1996	Cardoza	711/148
5,752,246	A		5/1998	Rogers et al.	
5,764,902	A	*	6/1998	Rothrock	345/753
5,812,773	A	*	9/1998	Norin	345/753
5,835,601	A	*	11/1998	Shimbo et al.	709/201
5,862,346	A	*	1/1999	Kley et al.	380/29
5,878,230	A		3/1999	Weber et al.	
5,924,103	A	*	7/1999	Ahmed et al	707/201
5,928,329	A		7/1999	Clark et al.	
5,943,676	A	*	8/1999	Boothby	707/201
5,974,238	A	*	10/1999	Chase, Jr.	709/248
5,999,932	A		12/1999	Paul	
5,999,947	A	*	12/1999	Zollinger et al	707/203
6,020,885	A	*	2/2000	Honda	345/757
6,021,427	A		2/2000	Spagna et al.	
6,023,700	A		2/2000	Owens et al.	
6,023,708	A	*	2/2000	Mendez et al.	707/203
6,034,621	A		3/2000	Kaufman	
6,073,165	A		6/2000	Narasimhan et al	
6,094,477	A		7/2000	Nada et al.	
6,108,709	A		8/2000	Shinomura et al.	
6,118,856	A		9/2000	Paarsmarkt et al.	
6,125,281	A		9/2000	Wells et al.	
6,131,096	A	*	10/2000	Ng et al.	707/10
6,131,116	A	*	10/2000	Riggins et al	709/219
6,138,146	A		10/2000	Moon et al.	
6,212,529	B1	*	4/2001	Boothby et al.	707/201
6,249,805	B1		6/2001	Fleming, III	
6,295,541	B1		9/2001	Bodnar et al.	
6,343,313	B1	*	1/2002	Salesky et al	345/751
6,510,455	B1		1/2003	Chen et al.	

OTHER PUBLICATIONS

Kohl, John T., et al.; "The Evolution of the *Kerberos* Authentication Service"; 1991; pp. 1–15; This paper is a revision of a paper presented at the Spring 1991 EurOpen Conference in Tromso, Norway.

Adams, Charlotte; "Multilevel Secure Networking Charges Ahead"; Federal Computer Week; Apr. 12, 1993; 5 pages.

Jaeger, Trent and Atul Prakash; "Implementation of a Discretionary Access Control Model for Script-based Systems"; IEEEE Jun. 1995; 15 pages

Research Disclosure; "Provide Auto–Forwarding Based on Criteria Selected by the User"; Oct. 1, 1989; 1 page; No. 306; Kenneth Mason Publications; XP000085405; ISSN; 0374–4353

* cited by examiner

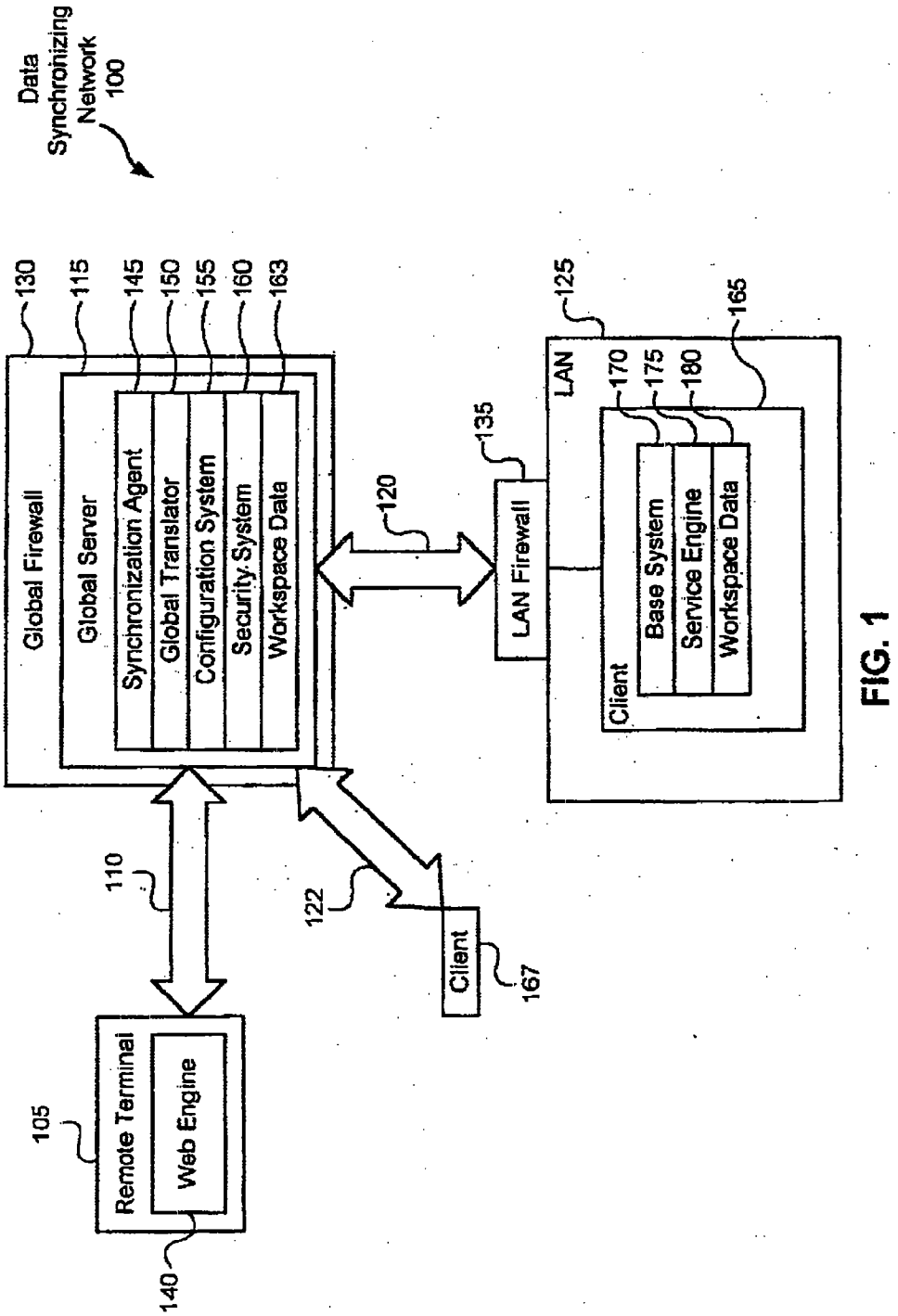

Data Synchronizing Network 100
130
115
145
150
155
160
163
Global Firewall
Global Server
Synchronization Agent
Global Translator
Configuration System
Security System
Workspace Data
120
135
LAN Firewall
125
LAN
165
170
175
180
Client
Base System
Service Engine
Workspace Data
110
122
Client
167
105
Remote Terminal
Web Engine
140

FIG. 1

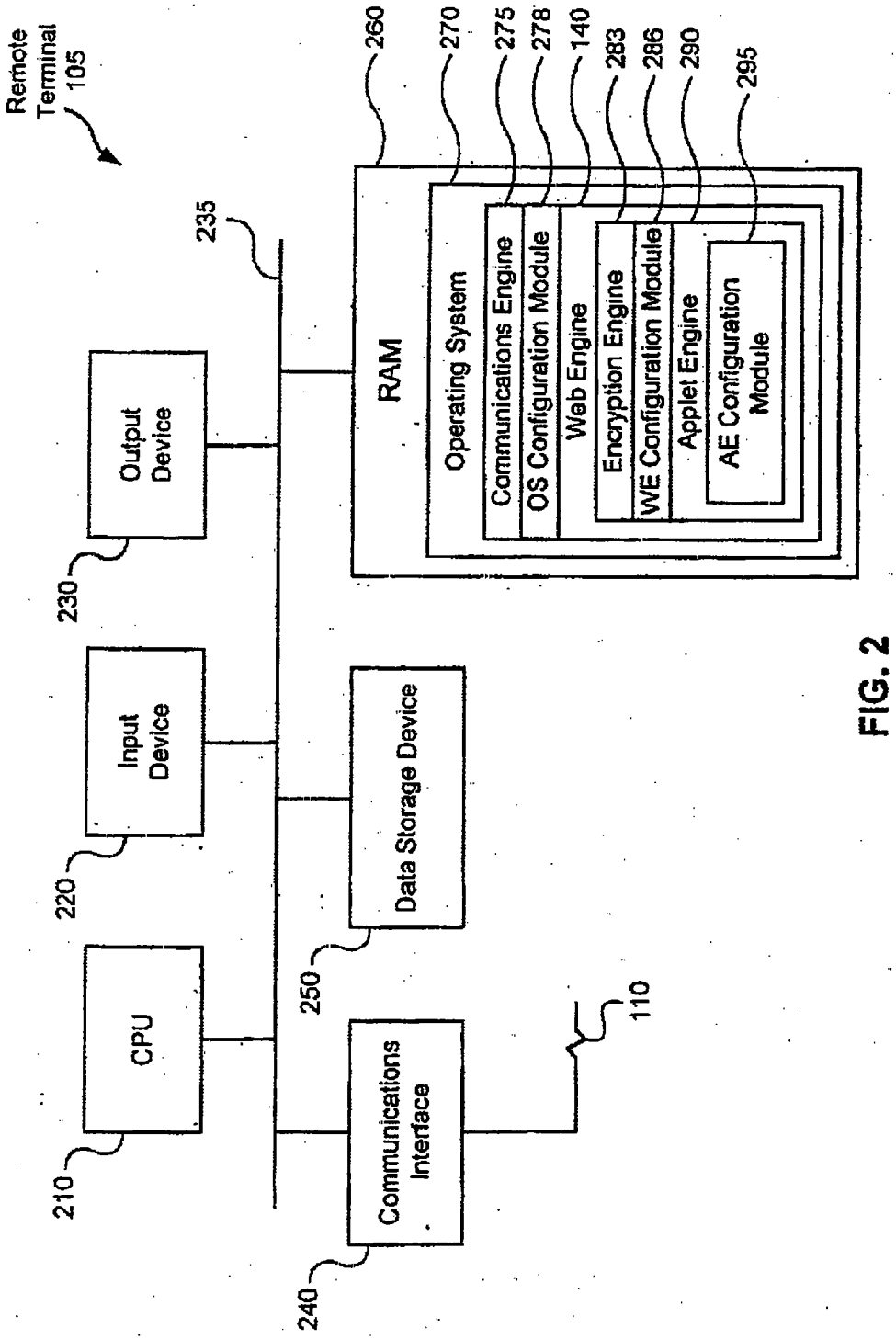

Remote Terminal 105
235
RAM
260
Operating System
270
Communications Engine
275
OS Configuration Module
278
Web Engine
140
Encryption Engine
283
WE Configuration Module
286
Applet Engine
290
AE Configuration Module
295
Output Device
230
Input Device
220
CPU
210
Data Storage Device
250
Communications Interface
240
110

FIG. 2

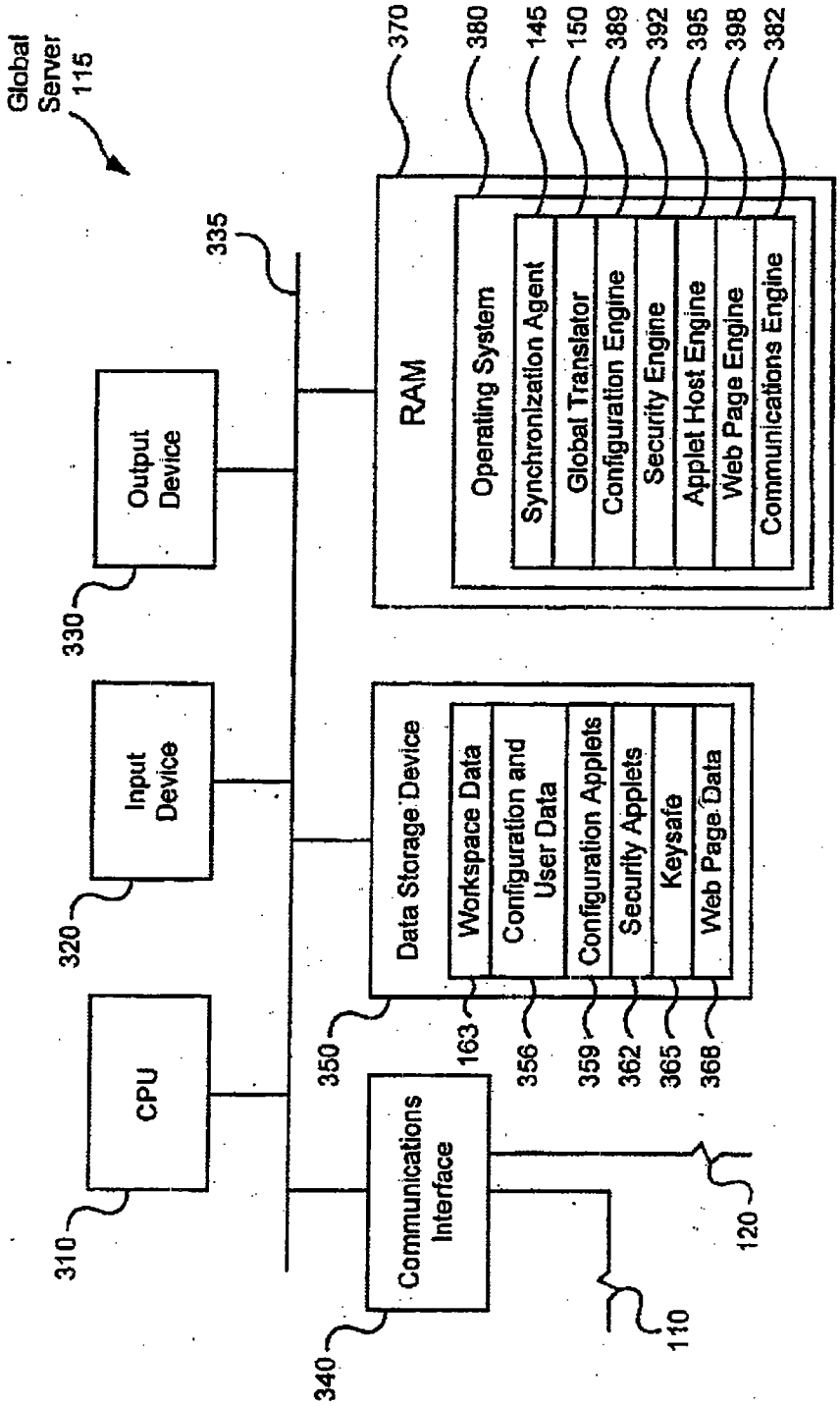

Global Server 115
370
380
145
150
389
392
395
398
382
335
RAM
Operating System
Synchronization Agent
Global Translator
Configuration Engine
Security Engine
Applet Host Engine
Web Page Engine
Communications Engine
Output Device
330
Input Device
320
CPU
310
Data Storage Device
Workspace Data
Configuration and User Data
Configuration Applets
Security Applets
Keysafe
Web Page Data
350
163
356
359
362
365
368
Communications Interface
340
110
120

FIG. 3

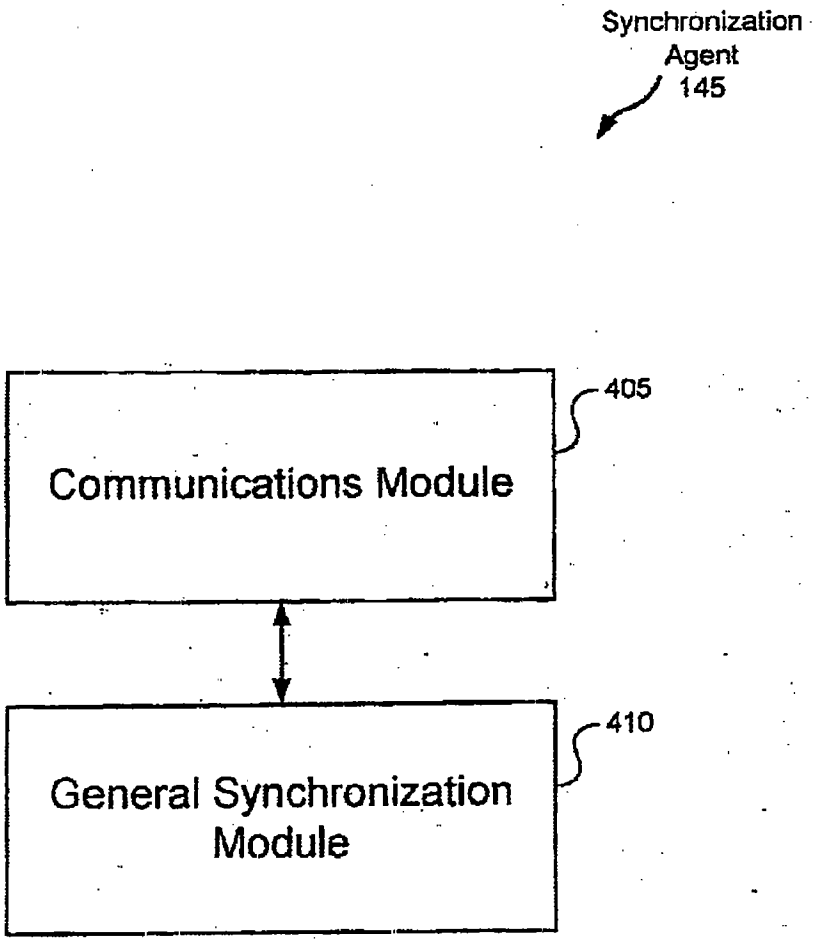

Synchronization Agent 145
Communications Module
405
General Synchronization Module
410

FIG. 4

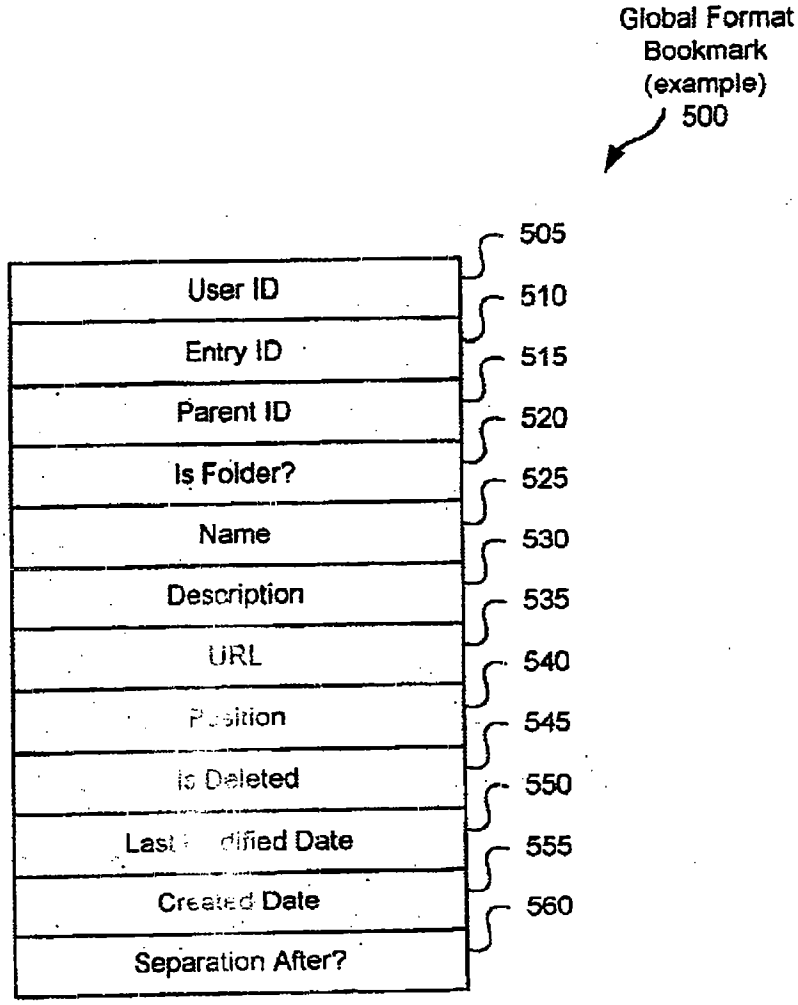

Global Format
Bookmark
(example)
500
505
User ID
510
Entry ID
515
Parent ID
520
Is Folder?
525
Name
530
Description
535
URL
540
Position
545
Is Deleted
550
Last Modified Date
555
Created Date
560
Separation After?

FIG. 5

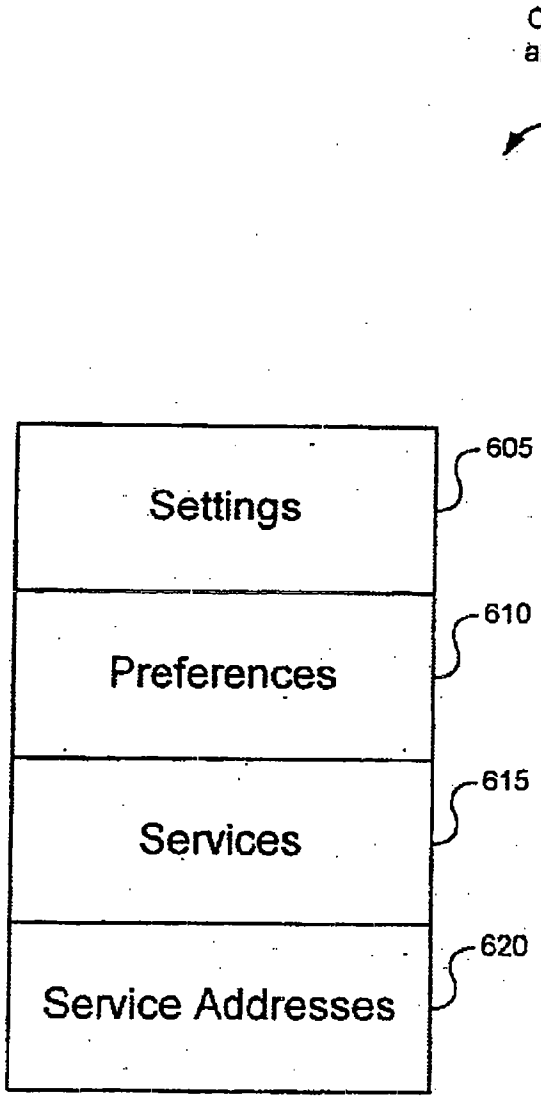

Configuration and user data
356
Settings
605
Preferences
610
Services
615
Service Addresses
620

FIG. 6

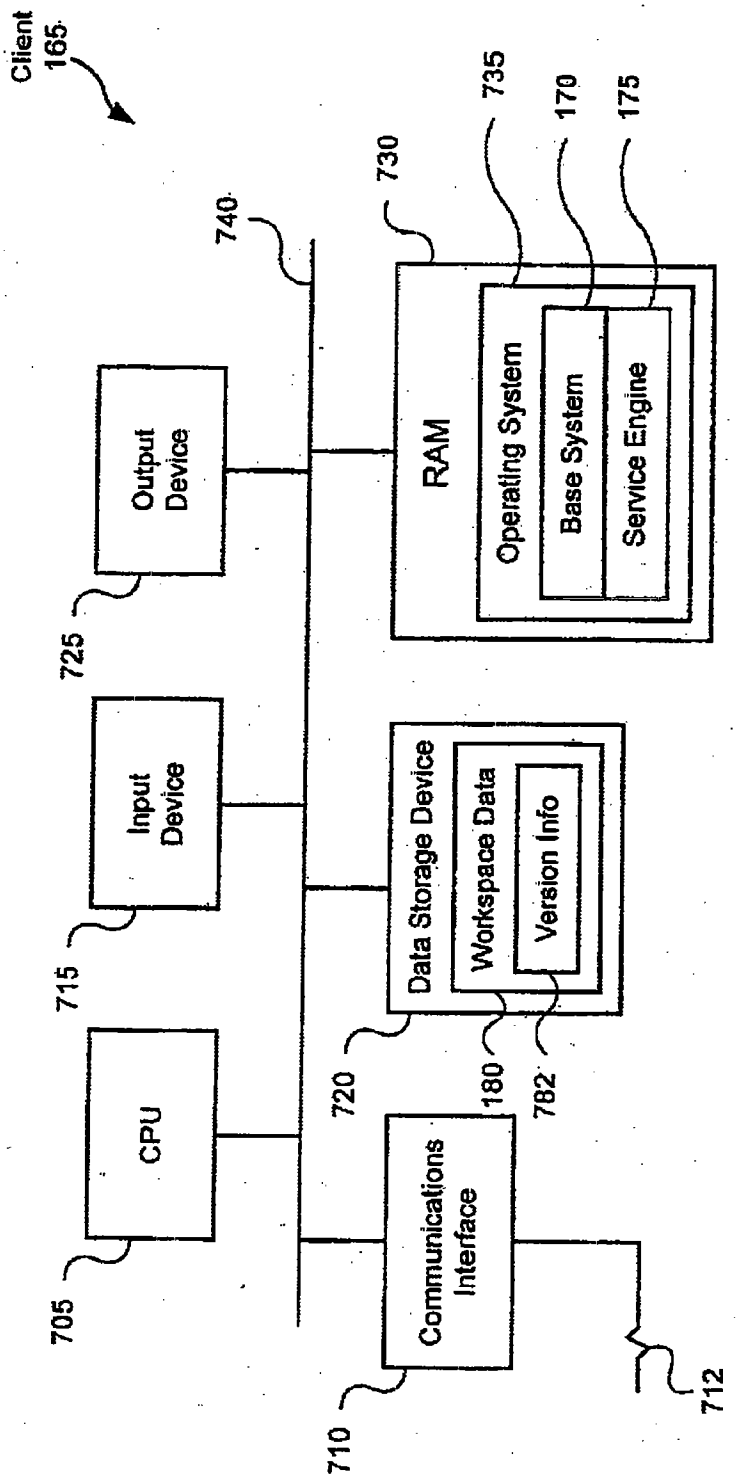

Client
165
740
730
735
170
175
RAM
Operating System
Base System
Service Engine
Output Device
725
Input Device
715
CPU
705
Data Storage Device
Workspace Data
Version Info
720
180
782
Communications Interface
710
712

FIG. 7

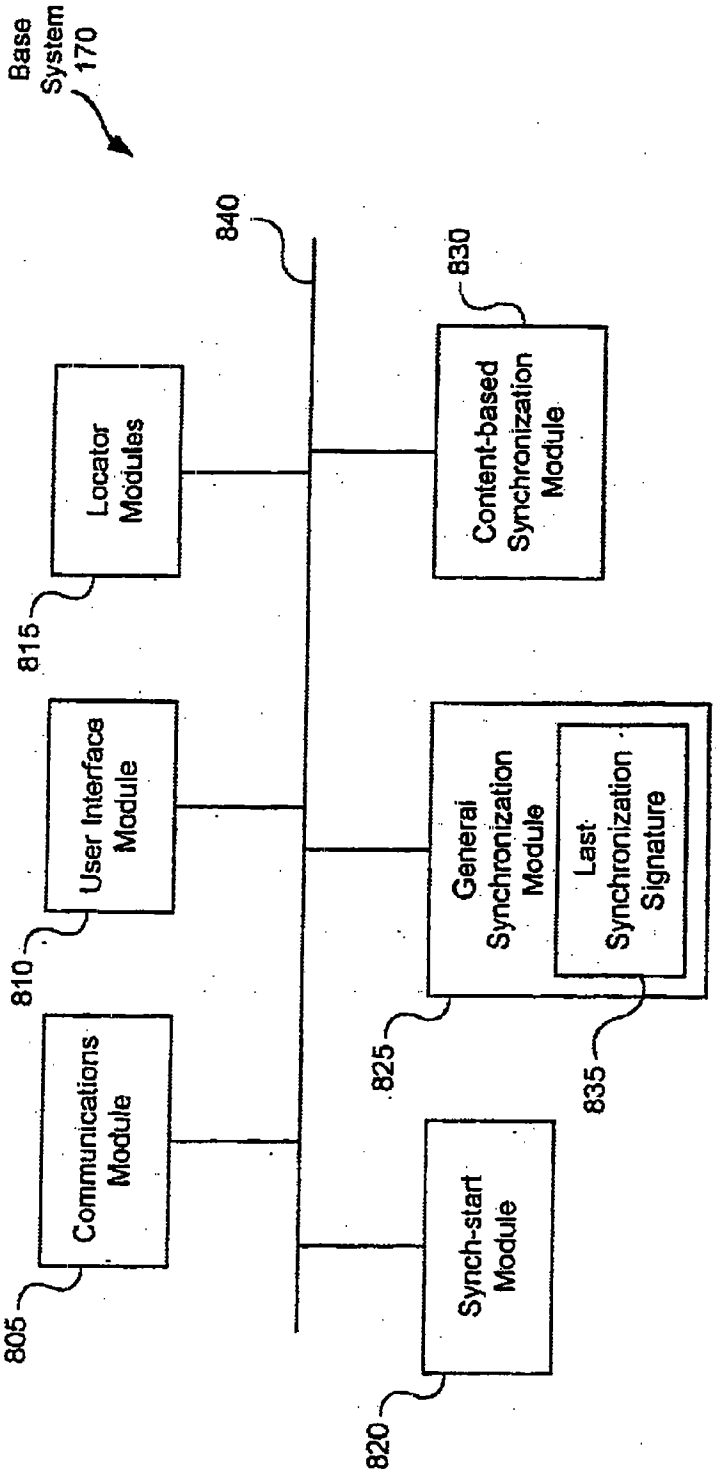

Base System 170
840
830
Content-based Synchronization Module
Locator Modules
815
User Interface Module
810
General Synchronization Module
Last Synchronization Signature
825
835
Communications Module
805
Synch-start Module
820

FIG. 8

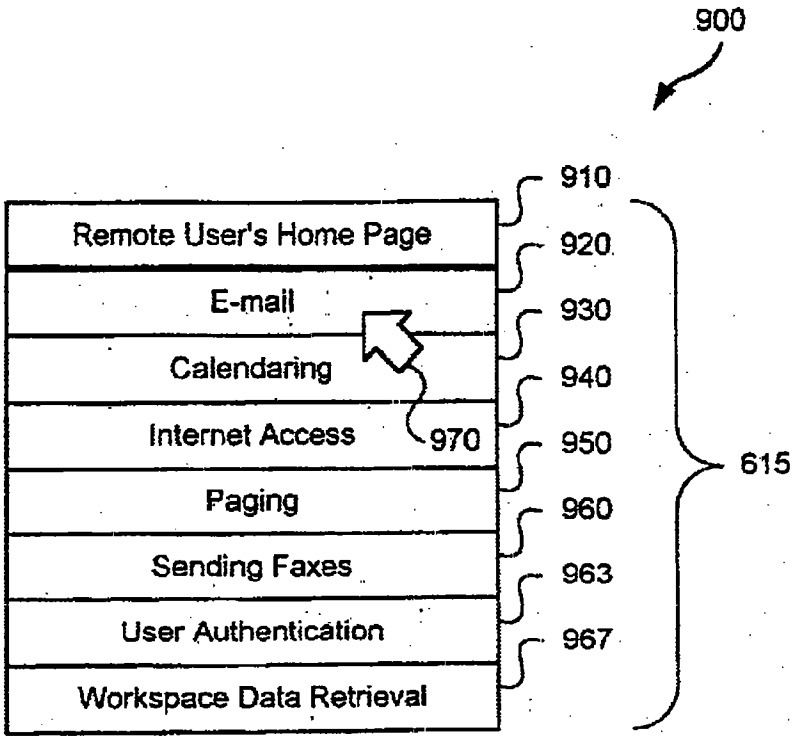

900
910
Remote User's Home Page
920
E-mail
930
Calendaring
940
Internet Access
970
950
Paging
960
Sending Faxes
963
User Authentication
967
Workspace Data Retrieval
615

FIG. 9

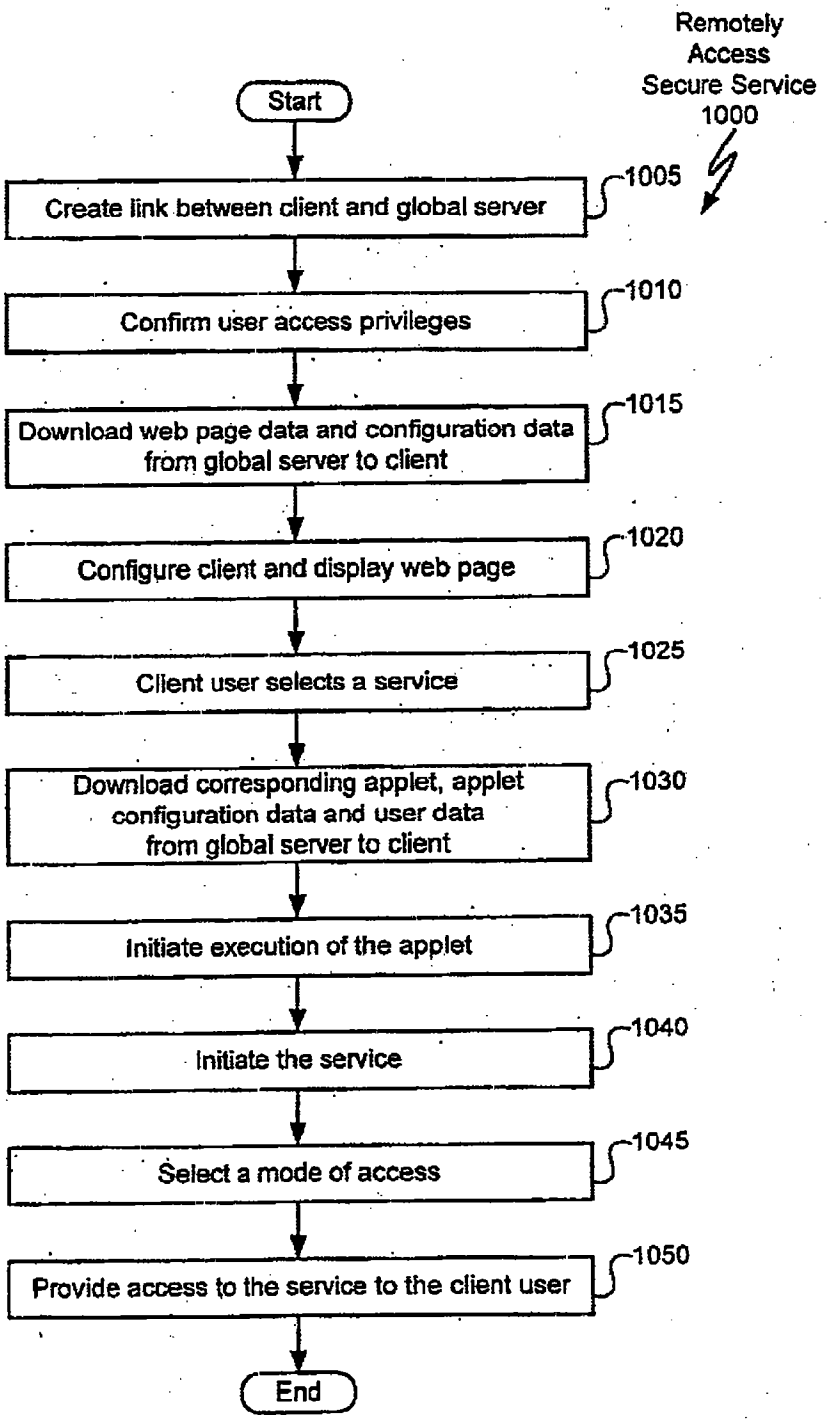

Remotely Access Secure Service 1000
Start
Create link between client and global server
1005
Confirm user access privileges
1010
Download web page data and configuration data from global server to client
1015
Configure client and display web page
1020
Client user selects a service
1025
Download corresponding applet, applet configuration data and user data from global server to client
1030
Initiate execution of the applet
1035
Initiate the service
1040
Select a mode of access
1045
Provide access to the service to the client user
1050
End

FIG. 10

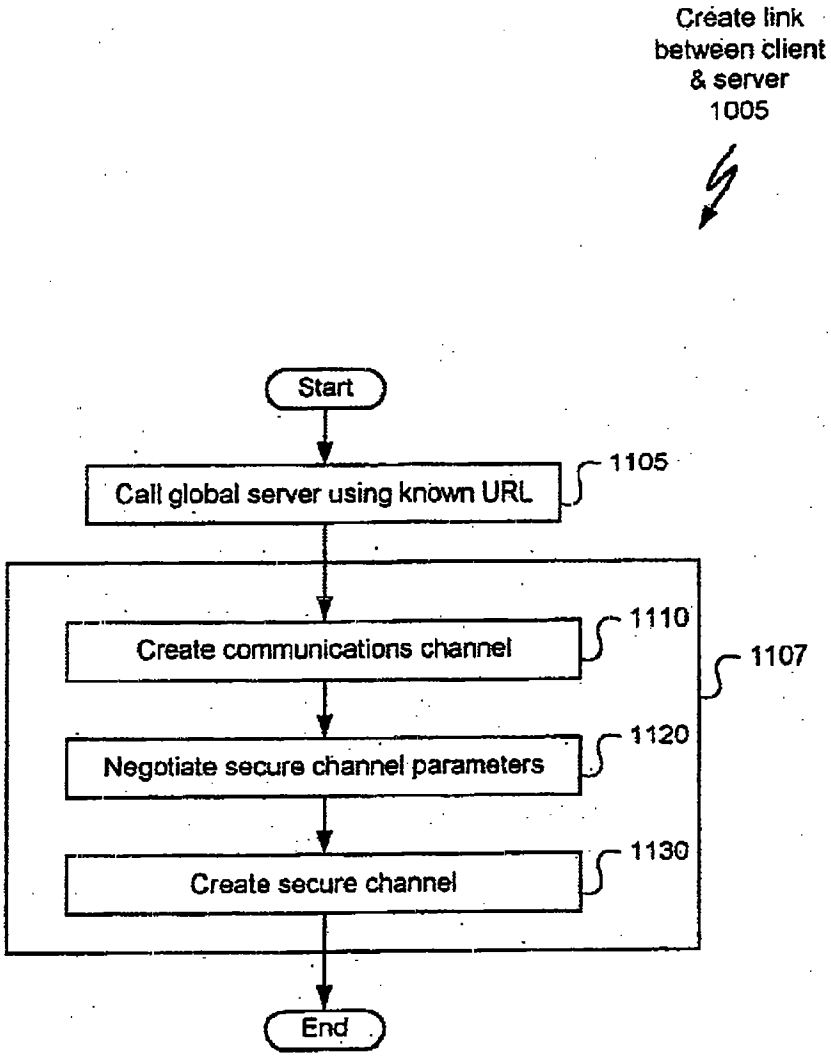

Create link between client & server 1005
Start
Call global server using known URL
1105
Create communications channel
1110
1107
Negotiate secure channel parameters
1120
Create secure channel
1130
End

FIG. 11

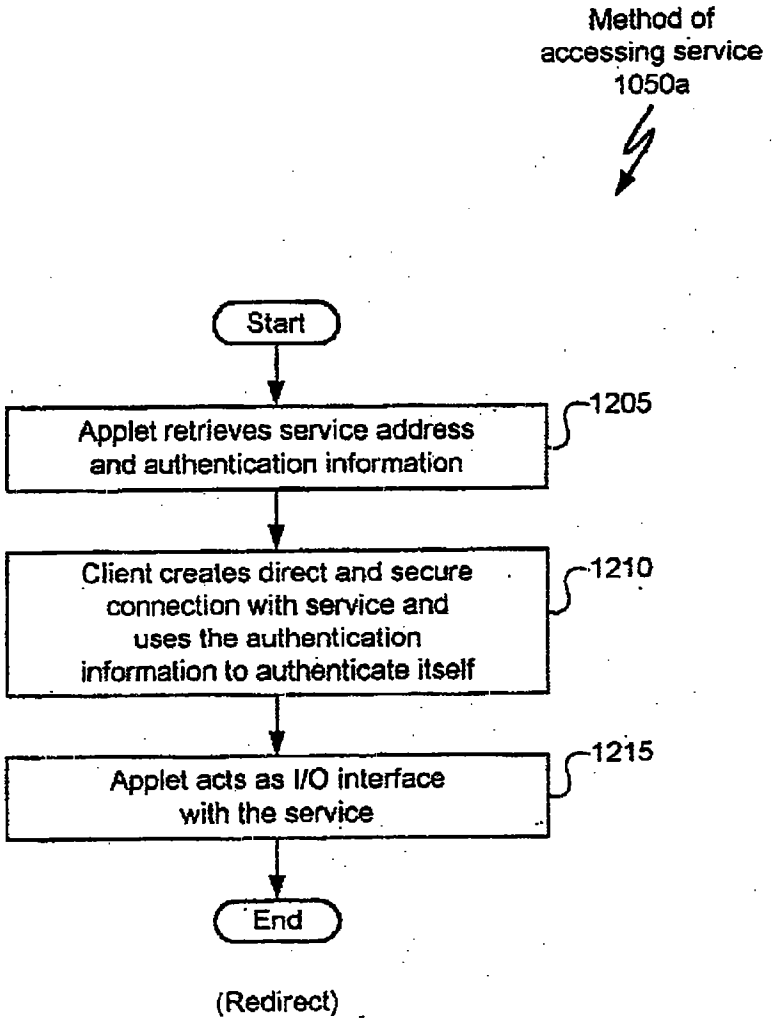

Method of accessing service 1050a
Start
Applet retrieves service address and authentication information
1205
Client creates direct and secure connection with service and uses the authentication information to authenticate itself
1210
Applet acts as I/O interface with the service
1215
End
(Redirect)

FIG. 12

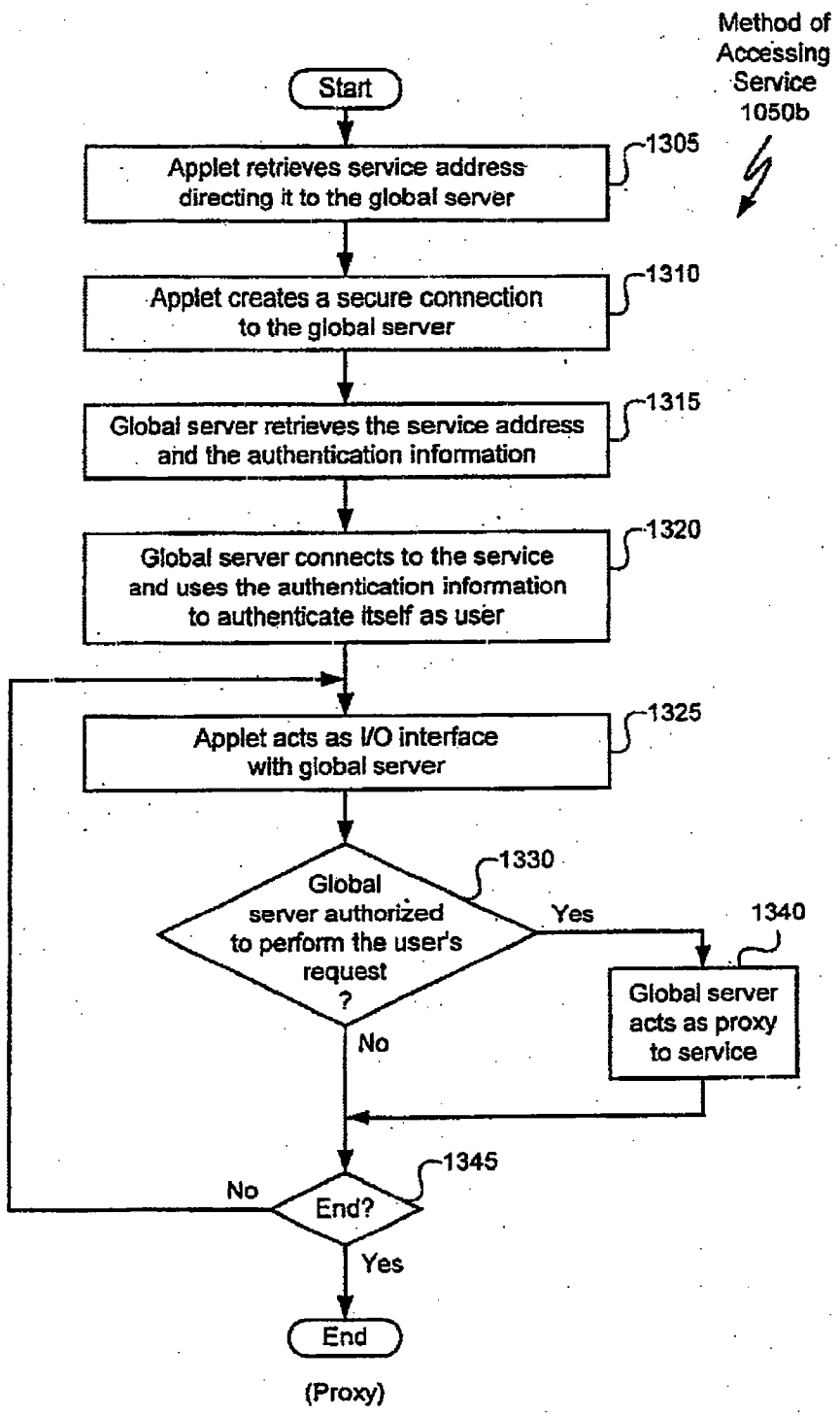

Method of Accessing Service 1050b
Start
Applet retrieves service address directing it to the global server
1305
Applet creates a secure connection to the global server
1310
Global server retrieves the service address and the authentication information
1315
Global server connects to the service and uses the authentication information to authenticate itself as user
1320
Applet acts as I/O interface with global server
1325
Global server authorized to perform the user's request ?
1330
Yes
1340
Global server acts as proxy to service
No
End?
1345
No
Yes
End
(Proxy)

FIG. 13

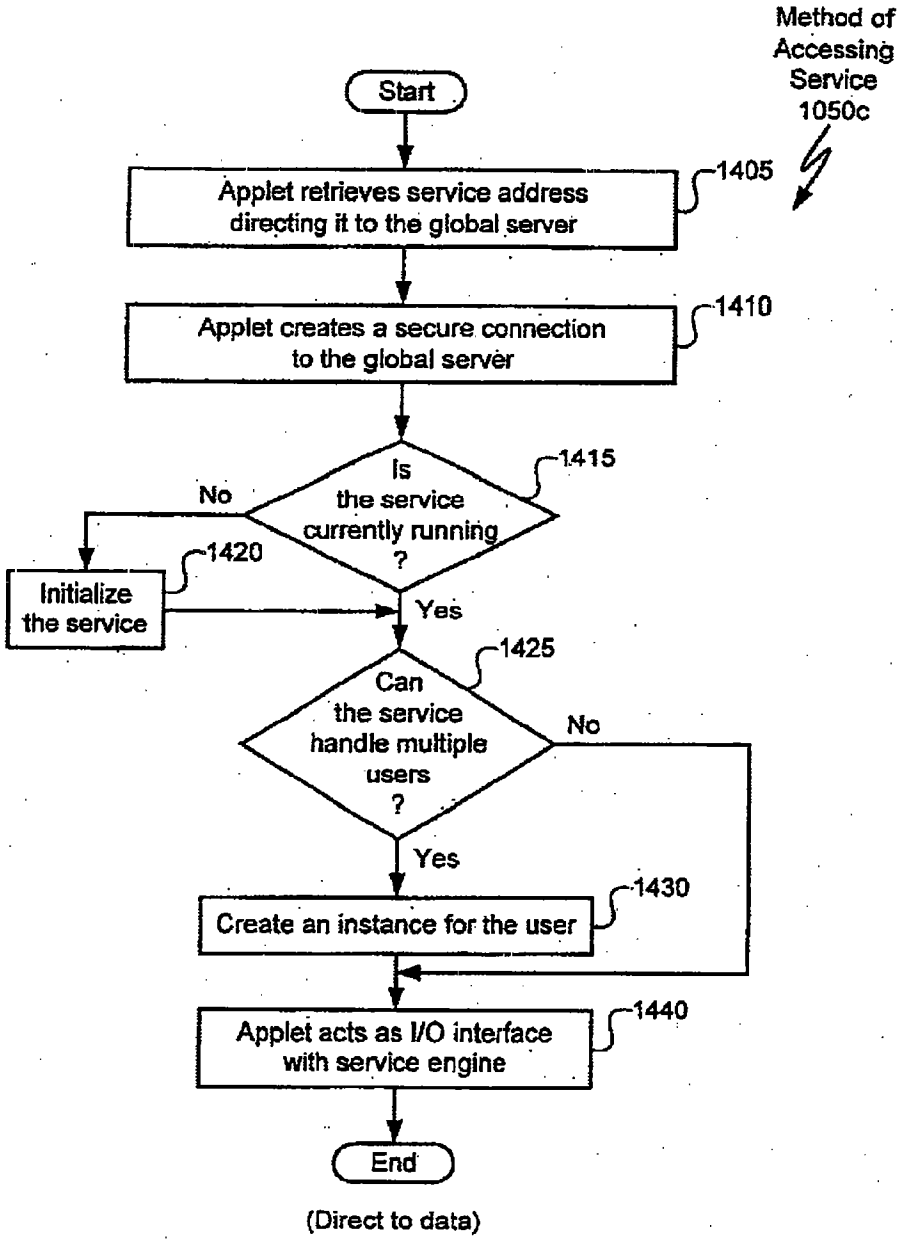

Method of Accessing Service 1050c
Start
Applet retrieves service address directing it to the global server
1405
Applet creates a secure connection to the global server
1410
Is the service currently running ?
1415
No
Initialize the service
1420
Yes
Can the service handle multiple users ?
1425
No
Yes
Create an instance for the user
1430
Applet acts as I/O interface with service engine
1440
End
(Direct to data)

FIG. 14

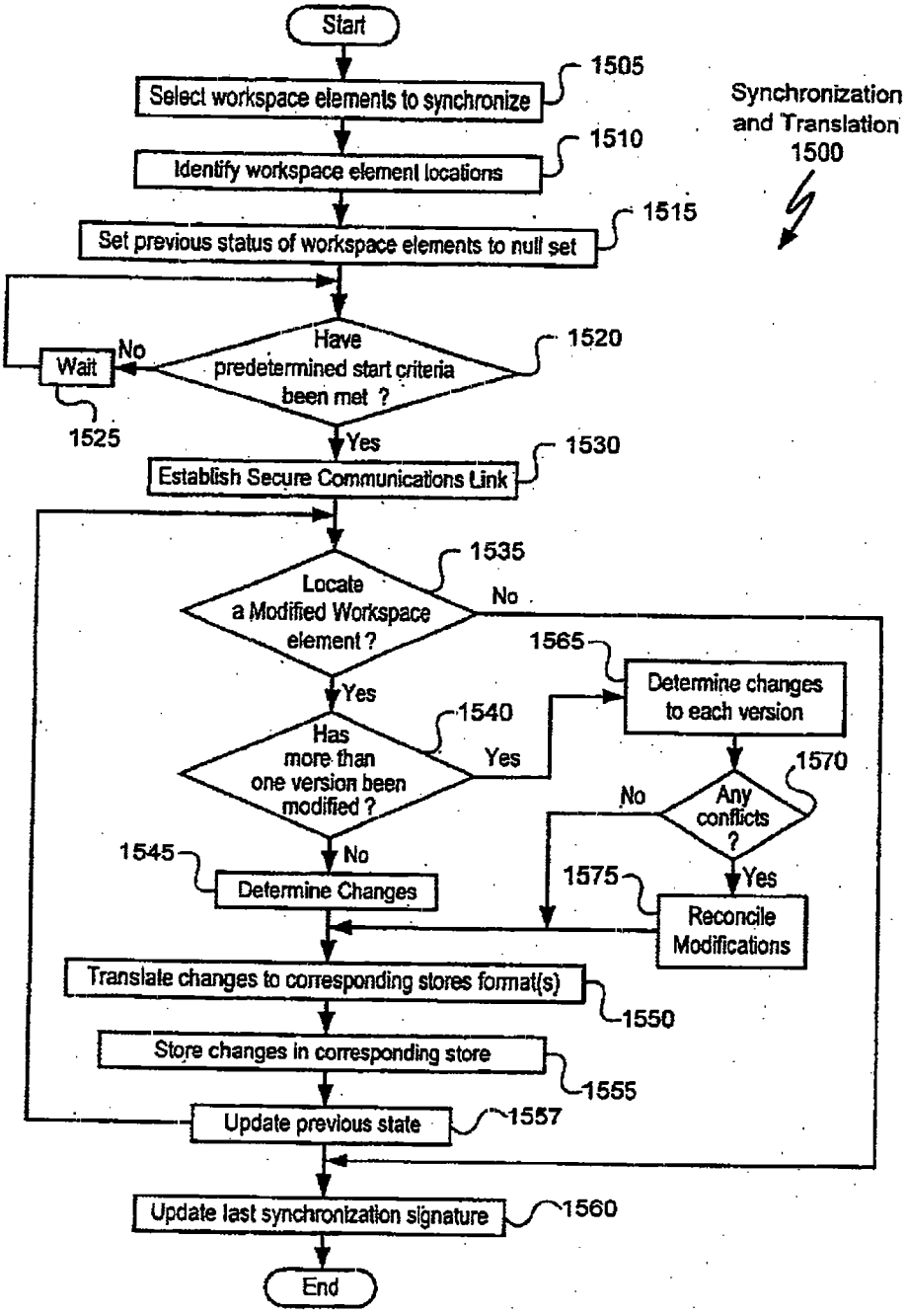

Start
Select workspace elements to synchronize
1505
Identify workspace element locations
1510
Set previous status of workspace elements to null set
1515
Synchronization and Translation
1500
Have predetermined start criteria been met ?
1520
No
Wait
1525
Yes
Establish Secure Communications Link
1530
Locate a Modified Workspace element ?
1535
No
Yes
Has more than one version been modified ?
1540
Yes
Determine changes to each version
1565
Any conflicts ?
1570
No
Yes
Reconcile Modifications
1575
No
Determine Changes
1545
Translate changes to corresponding stores format(s)
1550
Store changes in corresponding store
1555
Update previous state
1557
Update last synchronization signature
1560
End

FIG. 15

SYSTEM AND METHOD FOR GLOBALLY AND SECURELY ACCESSING UNIFIED INFORMATION IN A COMPUTER NETWORK

CROSS-REFERNCE TO RELATED APPLICATIONS

This application is a continuation and claims priority to and incorporates by reference U.S patent application Ser No. 08/903,118, entitled "System and Method For Globally and Securely Accessing Unified Information in a Computer Network" filed Jul. 30, 1997 of Daniel J Mendez, Mark D. Riggins, Prasad Wagle, Hong Q. Bui, Mason Ng, Sean Michael Quinlan, Christine C Ying, Christopher R. Zuleeg, David J. Cowan, Joanna A. Aptekar-Strober and R. Stanley Bailes, which is a continuation-in-part of U S. patent application Ser. No. 08/766,307, entitled "System and Method for Globally Accessing Computer Services," filed on Dec. 13, 1996 now U.S. Pat. No. 6,131,116 by inventors Mark D. Riggins, R. Stanley Bailes, Hong Q. Bui, David J. Cowan, Daniel J. Mendez, Mason Ng, Sean Michael Quinlan, Prasad Wagle, Christine C. Ying, Christopher R. Zuleeg and Joanna A. Aptekar-Strober; and of co-pending U S. patent application Ser No 08/841,950 entitled "System and Method for Enabling Secure Access to Services in a Computer Network," filed on Apr. 8, 1997 by inventor Mark Riggins; and of U.S. patent application Ser. No. 08/835.997 entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," filed on Apr 11, 1997 now U.S. Pat. No 6,085,192, by inventors Daniel J. Mendez, Mark J. Riggins, Prasad Wagle and Christine C. Ying; and of U.S. patent application Ser. No. 08/865,075 entitled "System and Method for Using a Global Translator to Synchronize Workspace Elements Across a Network," filed on May 29, 1997 now U.S. Pat No 6,023,708 by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying. These applications have been commonly assigned to RoamPage, Inc. and are incorporated herein by reference as if copied verbatim hereafter. Benefit of the earlier filing dates is claimed on all common subject matter

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for globally and securely accessing unified information in a computer network.

2. Description of the Background Art

The internet currently interconnects about 100,000 computer networks and several million computers. Each of these computers stores numerous application programs for providing numerous services, such as generating, sending and receiving e-mail, accessing World Wide Web sites, generating and receiving facsimile documents, storing and retrieving data, etc.

A roaming user, i.e., a user who travels and accesses a workstation remotely, is faced with several problems. Program designers have developed communication techniques for enabling the roaming user to establish a communications link and to download needed information and needed service application programs from the remote workstation to a local computer. Using these techniques, the roaming user can manipulate the data on the remote workstation and, when finished, can upload the manipulated data back from the remote workstation to the local computer. However, slow

computers and slow communication channels make downloading large files and programs a time-consuming process. Further, downloading files and programs across insecure channels severely threatens the integrity and confidentiality of the downloaded data

Data consistency is also a significant concern for the roaming user. For example, when maintaining multiple independently modifiable copies of a document, a user risks using an outdated version By the time the user notices an inconsistency, interparty miscommunication or data loss may have already resulted. The user must then spend more time attempting to reconcile the inconsistent versions and addressing any miscommunications.

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

Data consistency problems may also arise when using application programs from different vendors. For example, the Netscape Navigator™ web engine and the Internet Explorer™ web engine each store bookmarks for quick reference to interesting web sites. However, since each web engine uses different formats and stores bookmarks in different files, the bookmarks are not interchangeable In addition, one web engine may store a needed bookmark, and the other may not. A user who, for example, runs the Internet Explorer™ web engine at home and runs the Netscape Navigator™ web engine at work risks having inconsistent bookmarks at each location.

Therefore, a system and method are needed to enable multiple users to access computer services remotely without consuming excessive user time, without severely threatening the integrity and confidentiality of the data, and without compromising data consistency.

SUMMARY OF THE INVENTION

The present invention provides a system and methods for providing global and secure access to services and to unified (synchronized) workspace elements in a computer network. A user can gain access to a global server using any terminal, which is connected via a computer network such as the Internet to the global server and which is enabled with a web engine.

A client stores a first set of workspace data, and is coupled via a computer network to a global server. The client is configured to synchronize selected portions of the first set of workspace data (comprising workspace elements) with the global server, which stores independently modifiable copies of the selected portions. The global server may also store workspace data not received from the client, such as e-mail sent directly to the global server. Accordingly, the global server stores a second set of workspace data. The global server is configured to identify and authenticate a user attempting to access it from a remote terminal, and is configured to provide access based on the client configuration either to the first set of workspace data stored on the client or to the second set of workspace data stored on the global server. It will be appreciated that the global server can manage multiple clients and can synchronize workspace data between clients.

Service engines for managing services such as e-mail management, accessing bookmarks, calendaring, network

access, etc. may be stored anywhere in the computer network, including on the client, on the global server or on any other computer. The global server is configured to provide the user with access to services, which based on level of authentication management or user preferences may include only a subset of available services. Upon receiving a service request from the client, the global server sends configuration information to enable access to the service.

Each client includes a base system and the global server includes a synchronization agent. The base system and synchronization agent automatically establish a secure connection therebetween and synchronize the selected portions of the first set of workspace data stored on the client and the second set of workspace data stored on the global server. The base system operates on the client and examines the selected portions to determine whether any workspace elements have been modified since last synchronization. The synchronization agent operates on the global server and informs the base system whether any of the workspace elements in the second set have been modified. Modified version may then be exchanged so that an updated set of workspace elements may be stored at both locations, and so that the remote user can access an updated database. If a conflict exists between two versions, the base system then performs a responsive action such as examining content and generating a preferred version, which may be stored at both locations. The system may further include a synchronization-start module at the client site (which may be protected by a firewall) that initiates interconnection and synchronization when predetermined criteria have been satisfied.

A method of the present invention includes establishing a communications link between the client and the global server. The method includes establishing a communications link between the client and a service based upon user requests. The method receives configuration data and uses the configuration data to configure the client components such as the operating system, the web engine and other components. Configuring client components enables the client to communicate with the service and provides a user-and-service-specific user interface on the client. Establishing a communications link may also include confirming access privileges.

Another method uses a global translator to synchronize workspace elements. The method includes the steps of selecting workspace elements for synchronization, establishing a communications link between a client and a global server, examining version information for each of the workspace elements on the client and on the global server to determine workspace elements which have been modified since last synchronization. The method continues by comparing the corresponding versions and performing a responsive action. Responsive actions may include storing the preferred version at both stores or reconciling the versions using content-based analysis.

The system and methods of the present invention advantageously provide a secure globally accessible third party, i.e the global server. The system and methods provide a secure technique for enabling a user to access the global server and thus workspace data remotely and securely. Because of the global firewall and the identification and security services performed by the global server, corporations can store relatively secret information on the global server for use by authorized clients. Yet, the present invention also enables corporations to maintain only a portion of their secret information on the global server, so that there would be only limited loss should the global server be compromised. Further, the global server may advantageously act as a client proxy for controlling access to services, logging use of keys and logging access of resources.

A client user who maintains a work site, a home site, an off-site and the global server site can securely synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which control when the synchronization-start module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, the system is accessible using any connected terminal having a web engine such as an internet-enabled smart phone, television settop (e.g., web TV), etc. and is accessible over any distance. Since the system and method include format translation, merging of workspace elements between different application programs and different platforms is possible. Further, because synchronization is initiated from within the firewall, the typical firewall, which prevents in-bound communications and only some protocols of out-bound communications, does not act as an impediment to workspace element synchronization.

Further, a roaming user may be enabled to access workspace data from the global server or may be enabled to access a service for accessing workspace data from a client. For example, a user may prefer not to store personal information on the global server but may prefer to have remote access to the information. Further, the user may prefer to store highly confidential workspace elements on the client at work as added security should the global server be compromised.

The present invention may further benefit the roaming user who needs emergency access to information. The roaming user may request a Management Information Systems (MIS) director controlling the client to provide the global server with the proper keys to enable access to the information on the client. If only temporary access is desired, the keys can then be later destroyed either automatically or upon request. Alternatively, the MIS director may select the needed information as workspace elements to be synchronized and may request immediate synchronization with the global server. Accordingly, the global server and the client can synchronize the needed information, and the user can access the information from the global server after it has completed synchronization.

The present invention also enables the system and methods to synchronize keys, available services and corresponding service addresses to update accessibility of workspace data and services. For example, if the user of a client accesses a site on the Internet which requires a digital certificate and the user obtains the certificate, the system and methods of the present invention may synchronize this newly obtained certificate with the keys stored on the global server. Thus, the user need not contact the global server to provide it with the information. The synchronization means will synchronize the information automatically.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a secure data-synchronizing remotely accessible network in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of a FIG. 1 remote terminal;

FIG. 3 is a block diagram illustrating details of a FIG. 1 global server;

FIG. 4 is a block diagram illustrating details of a FIG 1 synchronization agent;

FIG. 5 is a graphical representation of an example bookmark in global format;

FIG. 6 is a graphical representation of the FIG. 3 configuration data;

FIG. 7 is a block diagram illustrating the details of a FIG. 1 client;

FIG 8 is a block diagram illustrating the details of a FIG. 1 base system;

FIG 9 illustrates an example services list;

FIG. 10 is a flowchart illustrating a method for remotely accessing a secure server;

FIG. 11 is a flowchart illustrating details of the FIG. 10 step of creating a link between a client and global server;

FIG. 12 is a flowchart illustrating details of the FIG. 10 step of providing access to a service in a first embodiment;

FIG. 13 is a flowchart illustrating details of the FIG. 10 step of providing access to a service in a second embodiment;

FIG. 14 is a flowchart illustrating details of the FIG. 10 step of providing access to a service in a third embodiment; and

FIG. 15 is a flowchart illustrating a method for synchronizing multiple copies of a workspace element over a secure network

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a network 100, comprising a first site such as a remote computer terminal 105 coupled via a communications channel 110 to a global server 115. The global server 115 is in turn coupled via a communications channel 120 to a second site such as a Local Area Network (LAN) 125 and via a communications channel 122 to a third site such as client 167. Communications channel 110, communications channel 120 and communications channel 122 may be referred to as components of a computer network such as the Internet. The global server 115 is protected by a global firewall 130, and the LAN 125 is protected by a LAN firewall 135

The LAN 125 comprises a client 165, which includes a base system 170 for synchronizing workspace data 180 (e-mail data, file data, calendar data, user data, etc.) with the global server 115 and may include a service engine 175 for providing computer services such as scheduling, e-mail, paging, word-processing or the like. Those skilled in the art will recognize that workspace data 180 may include other types of data such as application programs. It will be further appreciated that workspace data 180 may each be divided into workspace elements, wherein each workspace element may be identified by particular version information 782 (FIG. 7). For example, each e-mail, file, calendar, etc. may be referred to as "a workspace element in workspace data." For simplicity, each workspace element on the client 165 is referred to herein as being stored in format A. It will be further appreciated that the workspace data 180 or portions thereof may be stored at different locations such as locally on the client 165, on other systems in the LAN 125 or on other systems (not shown) connected to the global server 115

The client 167 is similar to the client 165. However, workspace data stored on the client 167 is referred to as being stored in format B, which may be the same as or different than format A. All aspects described above and below with reference to the client 165 are also possible with respect to the client 167. For example, client 167 may include services (not shown) accessible from remote terminal 105, may include a base system (not shown) for synchronizing workspace elements with the global server 115, etc.

The global server 115 includes a security system 160 for providing only an authorized user with secure access through firewalls to services. The security system 160 may perform identification and authentication services and may accordingly enable multiple levels of access based on the level of identification and authentication The global server 115 further includes a configuration system 155 that downloads configuration data 356 (FIGS. 3 and 6) to the remote terminal 105 to configure remote terminal 105 components such as the operating system 270 (FIG. 2), the web engine 283 (FIG 2), the applet engine 290 (FIG. 2), etc. The configuration system 155 uses the configuration data 356 to enable the remote terminal 105 to access the services provided by the service engine 175 and to provide a user-and-service-specific user interface.

The global server 115 stores workspace data 163, which includes an independently modifiable copy of each selected workspace element in the selected portions of the workspace data 180. Accordingly, the workspace data 163 includes an independently modifiable copy of each corresponding version information 782 (FIG. 7). The workspace data 163 may also include workspace elements which originate on the global server 115 such as e-mails sent directly to the global server 115 or workspace elements which are downloaded from another client (not shown). The global server 115 maintains the workspace data 163 in a format, referred to as a "global format," which is selected to be easily translatable by the global translator 150 to and from format A and to and from format B As with format A and format B, one skilled in the art knows that the global format actually includes a global format for each information type For example, there may be a global format for bookmarks (FIG. 5), a global format for files, a global format for calendar data, a global format for e-mails, etc.

The global server 115 also includes a synchronization agent 145 for examining the workspace elements of workspace data 163. More particularly, the base system 170 and the synchronization agent 145, collectively referred to herein as "synchronization means," cooperate to synchronize the workspace data 163 with the selected portions of the workspace data 180. The synchronization means may individually synchronize workspace elements (e.g., specific word processor documents) or may synchronize workspace element folders (e.g., a bookmark folder). Generally, the base system 170 manages the selected portions of the workspace data 180 within the LAN 125 and the synchronization agent 145 manages the selected portions of workspace data 163 within the global server 115 It will be appreciated that the global translator 150 cooperates with the synchronization means to translate between format A (or format B) and the global format It will be further appreciated that the global server 115 may synchronize the workspace data 163 with workspace data 180 and with the workspace data (not shown) on the client 167. Accordingly, the workspace data 163 can be easily synchronized with the workspace data (not shown) on the client 167.

The remote terminal 105 includes a web engine 140, which sends requests to the global server 115 and receives information to display from the global server 115. The web engine 140 may use HyperText Transfer Protocol (HTTP)

and HyperText Markup Language (HTML) to interface with the global server **115**. The web engine **140** may be enabled to run applets, which when executed operate as the security interface for providing access to the global server **115** and which operate as the application interface with the requested service. Using the present invention, a user can operate any remote client **105** connected to the Internet to access the global server **115**, and thus to access the services and the workspace data on or accessible by the global server **115**

FIG **2** is a block diagram illustrating details of the remote terminal **105**, which includes a Central Processing Unit (CPU) **210** such as a Motorola Power PC™ microprocessor or an Intel Pentium™ microprocessor. An input device **220** such as a keyboard and mouse, and an output device **230** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **235** to CPU **210**. A communications interface **240**, a data storage device **250** such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) **260** are further coupled via signal bus **235** to CPU **210**. The communications interface **240** is coupled to a communications channel **110** as shown in FIG. **1**.

An operating system **270** includes a program for controlling processing by CPU **210**, and is typically stored in data storage device **250** and loaded into RAM **260** (as shown) for execution Operating system **270** further includes a communications engine **275** for generating and transferring message packets via the communications interface **240** to and from the communications channel **110** Operating system **270** further includes an Operating System (OS) configuration module **278**, which configures the operating system **270** based on OS configuration data **356** (FIG. **3**) such as Transmission Control Protocol (TCP) data, Domain Name Server (DNS) addresses, etc. received from the global server **115**

Operating system **270** further includes the web engine **140** for communicating with the global server **115**. The web engine **140** may include a web engine (WE) configuration module **286** for configuring elements of the web engine **140** such as home page addresses, bookmarks, caching data, user preferences, etc. based on the configuration data **356** received from the global server **115** The web engine **140** may also include an encryption engine **283** for using encryption techniques to communicate with the global server **115**. The web engine **140** further may include an applet engine **290** for handling the execution of downloaded applets including applets for providing security. The applet engine **290** may include an Applet Engine (AE) configuration module **295** for configuring the elements of the applet engine **290** based on configuration data **356** received from the global server **115**.

FIG. **3** is a block diagram illustrating details of the global server **115**, which includes a Central Processing Unit (CPU) **310** such as a Motorola Power PC™ microprocessor or an Intel Pentium™ microprocessor. An input device **320** such as a keyboard and mouse, and an output device **330** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **335** to CPU **310** A communications interface **340**, a data storage device **350** such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) **370** are further coupled via signal bus **335** to CPU **310**. As shown in FIG **1**, the communications interface **340** is coupled to the communications channel **110** and to the communications channel **120**.

An operating system **380** includes a program for controlling processing by CPU **310**, and is typically stored in data storage device **350** and loaded into RAM **370** (as illustrated) for execution The operating system **380** further includes a communications engine **382** for generating and transferring message packets via the communications interface **340** to and from the communications channel **345**. The operating system **380** also includes a web page engine **398** for transmitting web page data **368** to the remote terminal **105**, so that the remote terminal **105** can display a web page **900** (FIG. **9**) listing functionality offered by the global server **115** Other web page data **368** may include information for displaying security method selections

The operating system **380** may include an applet host engine **395** for transmitting applets to the remote terminal **105**. A configuration engine **389** operates in conjunction with the applet host engine **395** for transmitting configuration applets **359** and configuration and user data **356** to the remote terminal **105**. The remote terminal **105**. executes the configuration applets **359** and uses the configuration and user data **356** to configure the elements (e.g., the operating system **270**, the web engine **140** and the applet engine **290**) of the remote terminal **105**. Configuration and user data **356** is described in greater detail with reference to FIG. **6**

The operating system **380** also includes the synchronization agent **145** described with reference to FIG. **1**. The synchronization agent **145** synchronizes the workspace data **163** on the global server **115** with the workspace data **180** on the client **165**. As stated above with reference to FIG. **1**, the global translator **150** translates between format A used by the client **165** and the global format used by the global server **115**.

The operating system **380** may also includes a security engine **392** for determining whether to instruct a communications engine **382** to create a secure communications link with a client **165** or terminal **105**, and for determining the access rights of the user. For example, the security engine **392** forwards to the client **165** or remote terminal **105** security applets **362**, which when executed by the receiver poll the user and respond back to the global server **115**. The global server **115** can examine the response to identify and authenticate the user

For example, when a client **165** attempts to access the global server **115**, the security engine **384** determines whether the global server **115** accepts in-bound communications from a particular port. If so, the security engine **392** allows the communications engine **382** to open a communications channel **345** to the client **165**. Otherwise, no channel will be opened. After a channel is opened, the security engine **392** forwards an authentication security applet **362** to the remote terminal **105** to poll the user for identification and authentication information such as for a user ID and a password. The authentication security applet **362** will generate and forward a response back to the global server **115**, which will use the information to verify the identity of the user and provide access accordingly.

It will be appreciated that a "request-servicing engine" may be the configuration engine **389** and the applet host engine **395** when providing services to a remote terminal **105** or client **165**. The request-servicing engine may be the web page engine **398** when performing workspace data **163** retrieval operations directly from the global server **115** The request-servicing engine may be the configuration engine **389** and the applet host engine **395** when performing workspace data **180** retrieval operations from the client **165** or from any other site connected to the global server **115**. The request-servicing engine may be security engine **392** when performing security services such as user identification and authentication. The request-servicing engine may be the

synchronization agent when the performing synchronization with the client **165**. Further, the request-servicing engine may be any combination of these components

FIG. **4** is a block diagram illustrating details of the synchronization agent **145**, which includes a communications module **405** and a general synchronization module **410**. The communications module **405** includes routines for compressing data and routines for communicating via the communications channel **120** with the base system **170**. The communications module **405** may further include routines for communicating securely channel through the global firewall **130** and through the LAN firewall **125**

The general synchronization module **410** includes routines for determining whether workspace elements have been synchronized and routines for forwarding to the base system **170** version information (not shown) of elements determined to be modified after last synchronization. The general synchronization module **410** may either maintain its own last synchronization signature (not shown), receive a copy of the last synchronization signature with the request to synchronize from the base system **170**, or any other means for insuring that the workspace data has been synchronized. The general synchronization module **410** further includes routines for receiving preferred versions of workspace data **180** workspace elements from the base system **170** and routines for forwarding preferred versions of workspace data **180** workspace elements to the base system **170**.

FIG. **5** illustrates an example bookmark workspace element in the global format The translator **150** incorporates all the information needed to translate between all incorporated formats. For example, if for a first client a bookmark in format A needs elements X, Y and Z and for a second client a bookmark in format B needs elements W, X and Y, the global translator **150** incorporates elements W, X, Y and Z to generate a bookmark in the global format. Further, the translator **150** incorporates the information which is needed by the synchronization means (as described below in FIG. **4**) such as the last modified date Accordingly, a bookmark in the Global Format may include a user identification (ID) **505**, an entry ID **510**, a parent ID **515**, a folder ID flag **520**, a name **525**, a description **530**, the Uniform Resource Locator (URL) **535**, the position **540**, a deleted ID flag **545**, a last modified date **550**, a created date **555** and a separation ID flag **560**.

FIG. **6** is a block diagram illustrating details of the configuration and user data **356**. Configuration data **356** includes settings **605** such as TCP data and the DNS address, web browser settings such as home page address, bookmarks and caching data, applet engine settings, and applet configuration data such as the user's e-mail address name and signature block. It will be appreciated that applet-specific configuration and user data **356** is needed, since the service may not be located on the user's own local client **165**. Configuration and user data **356** further includes predetermined user preferences **610** such as font, window size, text size, etc.

Configuration data **356** further includes the set of services **615**, which will be provided to the user Services **615** include a list of registered users and each user's list of user-preferred available services **615**. Services may also include a list of authentication levels needed to access the services **615**. Configuration and user data **137** further includes service addresses **620** specifying the location of each of the services **615** accessible via the global server **115**.

FIG. **7** is a block diagram illustrating details of the client **165**, which includes a CPU **705**, an input device **710**, an output device **725**, a communications interface **710**, a data storage device **720** and RAM **730**, each coupled to a signal bus **740**.

An operating system **735** includes a program for controlling processing by the CPU **705**, and is typically stored in the data storage device **720** and loaded into the RAM **730** (as illustrated) for execution. A service engine **175** includes a service program for managing workspace data **180** that includes version information (not shown). The service engine **175** may be also stored in the data storage device **720** and loaded into the RAM **730** (as illustrated) for execution. The workspace data **180** may be stored in the data storage device **330** As stated above with reference to FIG. **1**, the base system **170** operates to synchronize the workspace data **180** on the client **165** with the workspace data **163** on the global server **115**. The base system **170** may be also stored in the data storage device **720** and loaded into the RAM **730** (as shown) for execution. The base system **170** is described in greater detail with reference to FIG. **8**.

FIG. **8** is a block diagram illustrating details of the base system **170**, which includes a communications module **805**, a user interface module **810**, locator modules **815**, a synchronization-start ("synch-start") module **820**, a general synchronization module **825** and a content-based synchronization module **830**. For simplicity, each module is illustrated as communicating with one another via a signal bus **840**. It will be appreciated that the base system **170** includes the same components as included in the synchronization agent **145**

The communications module **805** includes routines for compressing data, and routines for communicating via the communications interface **710** (FIG. **7**) with the synchronization agent **145** (FIG **1**). The communications module **805** may include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the LAN firewall **135** and through the global firewall **130**. Because synchronization is initiated from within the LAN firewall **135** and uses commonly enabled protocols such as HyperText Transfer Protocol (HTTP), the typical firewall **135** which prevents in-bound communications in general and some outbound protocols does not act as an impediment to e-mail synchronization. Examples of communications modules **805** may include TCP/IP stacks or the AppleTalk™ protocol.

The user interface **810** includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface **810** operates in coordination with the client **165** components as described herein.

The locator modules **815** include routines for identifying the memory locations of the workspace elements in the workspace data **180** and the memory locations of the workspace elements in the workspace data **163**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query a user. It will be appreciated that the locator modules **815** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **115** to maintain updated memory locations of workspace elements

The synchronization-start module **820** includes routines for determining when to initiate synchronization of workspace data **163** and workspace data **180**. For example, the synchronization-start module **820** may initiate data synchro-

nization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria The synchronization-start module **820** initiates data synchronization by instructing the general synchronization module **825** to begin execution of its routines. It will be appreciated that communications with synchronization agent **145** preferably initiate from within the LAN **125**, because the typical LAN firewall **125** prevents in-bound communications and allows out-bound communications.

The general synchronization module **825** includes routines for requesting version information from the synchronization agent **145** (FIG. 1) and routines for comparing the version information against a last synchronization signature **835** such as a last synchronization date and time to determine which versions have been modified The general synchronization module **825** further includes routines for comparing the local and remote versions to determine if only one or both versions of a particular workspace element have been modified and routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include forwarding the modified version (as the preferred version) of a workspace element in workspace data **180** or forwarding just a compilation of the changes to the other store(s) Other appropriate synchronizing responsive actions may include, if reconciliation between two modified versions is needed, then instructing the content-based synchronization module **830** to execute its routines (described below).

It will be appreciated that the synchronization agent **145** preferably examines the local version information **124** and forwards only the elements that have been modified since the last synchronization signature **835**. This technique makes efficient use of processor power and avoids transferring unnecessary data across the communications channel **712**. The general synchronization module **825** in the LAN **135** accordingly compares the data elements to determine if reconciliation is needed. Upon completion of the data synchronization, the general synchronization module **825** updates the last synchronization signature **835**.

The content-based synchronization module **830** includes routines for reconciling two or more modified versions of workspace data **163**, **180** in the same workspace element. For example, if the original and the copy of a user workspace element have both been modified independently since the last synchronization, the content-based synchronization module **830** determines the appropriate responsive action. The content-based synchronization module **830** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i e , by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores. When both versions are stored at both stores, each version may include a link to the other version so that the user may be advised to select the preferred version.

It will be appreciated that any client **165** that wants synchronization may have a base system **170**. Alternatively, one base system **170** can manage multiple clients **165**. It will be further appreciated that for a thin client **165** of limited computing power such as a smart telephone, all synchronization may be performed by the global server **115** Accordingly, components of the base system **170** such as the user interface module **810**, the locator modules **815**, the general synchronization module **825** and the content-based synchronization module **830** may be located on the global server **115**. To initiate synchronization from the client **165**, the client **165** includes the communications module **805** and the synch-start module **820**

FIG **9** illustrates an example list **900** of accessible services provided by a URL-addressable HyperText Markup Language (HTML)-based web page, as maintained by the web page engine **398** of the global server **115**. The list **900** includes a title **910** "Remote User's Home Page," a listing of the provided services **615** and a pointer **970** for selecting one of the provided services **615** As illustrated, the provided services may include an e-mail service **920**, a calendaring service **930**, an internet access service **940**, a paging service **950**, a fax sending service **960**, a user authentication service **963** and a workspace data retrieval service **967**. Although not shown, other services **615** such as bookmarking, QuickCard™, etc. may be included in the list **900**. Although the web page provides the services **615** in a list **900**, other data structures such as a pie chart or table may alternatively be used.

FIG. **10** is a flowchart illustrating a method **1000** for enabling a user to access the services **615** in the computer network system **100**. Method **1000** begins by the remote terminal **105** in step **1005** creating a communications link with the global server **115**. The global server **115** in step **1010** confirms that the user has privileges to access the functionality of the global server **115**. Confirming user access privileges may include examining a user certificate, obtaining a secret password, using digital signature technology, performing a challenge/response technique, etc. It will be appreciated that the security engine **392** may cause the applet host engine **395** to forward via the communications channel **345** to the remote terminal **105** an authentication security applet **362** which when executed communicates with the global server **115** to authenticate the user.

After user access privileges are confirmed, the web page engine **398** of the global server **115** in step **1015** transmits web page data **368** and configuration and user data **356** to the remote terminal **105**. The web engine **140** of the remote terminal **105** in step **1020** uses the web page data **368** and the configuration and user data **356** to display a web page service list **900** (FIG. 9) on the output device **230**, and to enable access to the services **615** which the global server **115** offers An example service list **900** is shown and described with reference to FIG **9** Configuration of the remote terminal **105** and of the web page **700** is described in detail in the cross-referenced patent applications

From the options listed on the web page **900**, the user in step **1025** selects a service **615** via input device **220** In response, the request-servicing engine (described with reference to FIG. 3) provides the selected service **615**. For example, the applet host engine **395** of the global server **115** in step **1030** may download to the remote terminal **105** a corresponding applet **359** and configuration and user data **356** for executing the requested service **615** Alternatively, the web page engine **398** may use, for example, HTTP and HTML to provide the selected service **615** As described above with reference to FIG. **6**, the configuration and user data **356** may include user-specific preferences such as user-preferred fonts for configuring the selected service **615**. Configuration and user data **356** may also include user-specific and service-specific information such as stored bookmarks, calendar data, pager numbers, etc Alternatively, the corresponding applet **359** and the configuration and user data **356** could have been downloaded in step **1015**. Providing access to the service by an applet **359** is described in greater detail below with reference to FIGS. **12–14**.

The applet engine **290** of the remote terminal **105** in step **1035** initiates execution of the corresponding downloaded

applet The global server **115** in step **1040** initiates the selected service **615** and in step **1045** selects one of three modes described with reference to FIGS. 12–14 for accessing the service **615**. For example, if the user selects a service **615** on a service server (e.g., the client **165**) that is not protected by a separate firewall, then the global server **115** may provide the user with direct access. If the user selects a service **615** provided by a service server within the LAN **125**, then the global server **115** may access the service **615** as a proxy for the user. It will be appreciated that each firewall **130** and **135** may store policies establishing the proper mode of access the global server **115** should select. Other factors for selecting mode of access may include user preference, availability and feasibility. The global server **115** in step **1050** uses the selected mode to provide the remote terminal **105** user with access to the selected service **615**.

FIG. **11** is a flowchart illustrating details of step **1005**, which begins by the remote terminal **105** in step **1105** using a known Uniform Resource Locator (URL) to call the global server **115**. The global server **115** and the remote terminal **105** in step **1107** create a secure communications channel therebetween, possibly by applying Secure Sockets Layer (SSL) technology. That is, the security engine **392** of the global server **115** in step **1110** determines if in-bound secure communications are permitted and, if so, creates a communications channel with the remote terminal **105**. The web engine **140** of the remote terminal **105** and the security engine **392** of the global server **115** in step **1115** negotiate secure communications channel parameters, possibly using public key certificates. An example secure communications channel is RSA with RC4 encryption. Step **1115** thus may include selecting an encryption protocol which is known by both the global server **115** and the remote terminal **105**. The encryption engine **283** of the remote terminal **105** and secure communications engine **392** of the global server **115** in step **1120** use the secure channel parameters to create the secure communications channel. Method **505** then ends.

FIG. **12** is a flowchart illustrating details of step **1050** in a first embodiment, referred to as step **1050***a*, wherein the global server **115** provides the remote terminal **105** with a direct connection to a service **615**. Step **1050***a* begins by the applet engine **290** in step **1205** running a configuration applet **359** for the selected service **615** that retrieves the service address **620** from data storage device **380** and the authentication information from the keysafe **365**. The communications interface **340** in step **1210** creates a direct and secure connection with the communications interface **340** of the global server **115** at the retrieved service address **620**, and uses the authentication information to authenticate itself. The applet in step **1215** acts as the I/O interface with the service **615**. Step **1050***a* then ends.

FIG. **13** is a flowchart illustrating details of step **1050** in a second embodiment, referred to as step **1050***b*, wherein the global server **115** acts for the remote terminal **105** as a proxy to the service **615**. Step **1050***b* begins with a configuration applet **359** in step **1305** requesting the service address **620** for the selected service **615**, which results in retrieving the service address **620** directing the applet **359** to the global server **115**. The applet **359** in step **1310** creates a connection with communications interface **340** of the global server **115**. The global server **115** in step **1315** retrieves the service address **620** of the selected service **615** and the authentication information for the selected service **615** from the keysafe **365**. The communications interface **340** of the global server **115** in step **1320** negotiates secure channel parameters for creating a secure channel with the service server **1014**. The communications interface **340** in step **1320** also authenticates itself as the user

Thereafter, the applet **359** in step **1325** acts as the I/O interface with the communications interface **340** of the global server **115**. If the global server **115** in step **1330** determines that it is unauthorized to perform a remote terminal **105** user's request, then the global server **115** in step **1345** determines whether the method **1050***b* ends, e.g., whether the user has quit. If so, then method **1050***b* ends. Otherwise, method **1050***b* returns to step **1325** to obtain another request. If the global server **115** in step **1330** determines that it is authorized to perform the remote terminal **105** user's request, then the global server **115** in step **1340** acts as the proxy for the remote terminal **105** to the service **615**. As proxy, the global server **115** forwards the service request to the selected service **615** and forwards responses to the requesting applet **359** currently executing on the remote terminal **105**. Method **1050***b* then jumps to step **1345**.

FIG. **14** is a flowchart illustrating details of step **1050** in a third embodiment, referred to as step **1050***c*, wherein the service **615** being requested is located on the global server **115**. Step **1050** begins with an applet in step **1405** retrieving the service address **620** for the selected service **615**, which results in providing the configuration applet **359** with the service address **620** of the service **615** on the global server **115**. Thus, the applet in step **1410** creates a secure connection with the global server **115**. No additional step of identification and authentication is needed since the remote terminal **105** has already identified and authenticated itself to the global server **115** as described with reference to step **1010** of FIG. **10**.

In step **1415**, a determination is made whether the service **615** is currently running. If so, then in step **1425** a determination is made whether the service **615** can handle multiple users. If so, then the global server **115** in step **1430** creates an instance for the user, and the applet in step **1440** acts as the I/O interface with the service **615** on the global server **115**. Method **1050***c* then ends. Otherwise, if the service **615** in step **1425** determines that it cannot handle multiple users, then method **1050***c* proceeds to step **1440**. Further, if in step **1415** the global server **115** determines that the service **615** is not currently running, then the global server **115** in step **1420** initializes the service **615** and proceeds to step **1425**.

FIG. **15** is a flowchart illustrating a method **1500** for using a global translator **150** to synchronize workspace data **163** and workspace data **180** in a secure network **100**. Method **1500** begins with the user interface **900** in step **1505** enabling a user to select workspace elements of workspace data **163** and workspace data **180** for the synchronization means to synchronize. The locator modules **815** in step **1510** identify the memory locations of the workspace elements in workspace data **163** and workspace data **180**. If a selected workspace element does not have a corresponding memory location, such as in the case of adding new workspace elements to the global server **115**, then one is selected. The selected memory location may be a preexisting workspace element or a new workspace element. As stated above, workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The general synchronization module **825** in step **1515** sets the previous status of the workspace elements equal to the null set, which indicates that all information of the workspace element has been added.

The synchronization-start module **820** in step **1520** determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step **1505** should start. If not, then the

synchronization-start module **820** in step **1525** waits and loops back to step **1520**. Otherwise, the communications module **805** and the communications module **405** in step **1530** establish a secure communications channel therebetween.

The general synchronization module **825** in step **1535** determines whether any workspace elements have been modified. That is, the general synchronization module **825** in step **1535** examines the version information of each selected workspace element in the workspace data **180** against the last synchronization signature **435** to locate modified workspace elements. This comparison may include comparing the date of last modification with the date of last synchronization, or may include a comparison between the current status and the previous status as of the last interaction. Similarly, the general synchronization module **815** examines the version information of each corresponding workspace element in workspace data **163** and the last synchronization signature **435** to locate modified workspace elements.

If in step **1535** no modified workspace elements or folders are located, then the general synchronization module **825** in step **1560** updates the last synchronization signature **435** and method **1500** ends. Otherwise, the general synchronization module **825** in step **1540** determines whether more than one version of a workspace element has been modified since the last synchronization.

If only one version has been modified, then the corresponding general synchronization module **825** in step **1545** determines the changes made. As stated above, determining the changes made may be implemented by comparing the current status of the workspace element against the previous status of the workspace element as of the last interaction therebetween. If the changes were made only to the version in the workspace data **163**, then the global translator **150** in step **1550** translates the changes to the format used by the other store, and the general synchronization module **410** in step **1555** forwards the translated changes to the general synchronization module **825** for updating the outdated workspace element in the workspace data **180**. If the updated version is a workspace element in the workspace data **180**, then the general synchronization module **825** sends the changes to the updated version to the global translator **150** for translation and then to the general synchronization module **410** for updating the outdated workspace element in the workspace data **163**. The general synchronization module **825** and the general synchronization module **410** in step **1557** update the previous state of the workspace element to reflect the current state as of this interaction. Method **1500** then returns to step **1535**.

If the general synchronization module **825** in step **1540** determines that multiple versions have been modified, then the general synchronization module **825** in step **1565** computes the changes to each version and in step **1570** instructs the content-based synchronization module **830** to examine content to determine if any conflicts exist. For example, the content-based synchronization module **830** may determine that a conflict exists if a user deletes a paragraph in one version and modified the same paragraph in another version. The content-based synchronization module **830** may determine that a conflict does not exist if a user deletes different paragraphs in each version. If no conflict is found, then method **1500** jumps to step **1550** for translating and forwarding the changes in each version to the other store. However, if a conflict is found, then the content-based synchronization module **830** in step **1575** reconciles the modified versions. As stated above, reconciliation may include requesting instructions from the user or based on previously selected preferences performing responsive actions such as storing both versions at both stores. It will be appreciated that a link between two versions may be placed in each of the two versions, so that the user will recognize to examine both versions to select the preferred version. Method **1500** then proceeds to step **1550**.

It will be further appreciated that in step **1510** new workspace elements and preexisting workspace elements to which new workspace elements will be merged are set to "modified" and the previous status is set to the null set. Thus, the general synchronization module **825** in step **1540** will determine that more that one version has been modified and the content-based synchronization module **830** in step **1570** will determine that no conflict exists. The changes in each will be translated and forwarded to the other store. Accordingly, the two versions will be effectively merged and stored at each store.

For example, if a first bookmark folder was created by the web engine **140** on the client **165**, a second folder was created by a web engine **140** on the remote terminal **105**, no preexisting folder existed on the global server **115** and the user selected each of these folders for synchronization, then the synchronization means will effectively merge the first and second folders. That is, the general synchronization module **825** on the client **165** will determine that the first folder has been modified and the previous status is equal to the null set. The general synchronization module **825** will determine and send the changes, i.e., all the workspace elements in the first folder, to a new global folder on the global server **115**. Similarly, the general synchronization module (not shown) on the remote terminal **105** will determine that, as of its last interaction, the previous status of each of the second and the global folders is the null set. The general synchronization module **825** will instruct the content-based synchronization module **830** to examine the changes made to each folder to determine whether a conflict exists. Since no conflicts will exist, the general synchronization module **825** will forward the changes to the global folder and the general synchronization module **410** will forward its changes to the second store, thereby merging the workspace elements of the first and second folders in the global and second folders. The general synchronization module **410** will inform the general synchronization module **825** that the global folder has been modified relative to the last interaction, and will forward the new changes to the first folder. Thus, the first and second folders will be merged and stored at each store.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, a server can be any computer which is polled by a client. Thus, the remote terminal **105** may be referred to as a type of client. Although the system and method have been described with reference to applets, other downloadable executables such as Java™ applets, Java™ applications or ActiveX™ control developed by the Microsoft Corporation can alternatively be used. Components of this invention may be implemented using a programmed general-purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The invention is limited only by the following claims.

What is claimed is:

1. A method for synchronizing workspace data, comprising:

storing first workspace data on a first device;

storing second workspace data on a second device;

determining differences between the first workspace data and the second workspace data;

storing the differences at a global server; and

sending the differences from the global server to the second device.

2. The method of claim 1, wherein the first workspace data comprise a workspace data element from a first user of the first device to a second user of the second device.

3. The method of claim 2, wherein the workspace data element includes data selected from a group including email data, file data, calendar data, user data and bookmark data.

4. The method of claim 1, wherein at least one of the first device and the second device is selected from a group including a smart phone, a television settop box and a personal computer

5. The method of claim 1, further comprising continuing to store the differences at the global server is continued after the sending.

6. The method of claim 1, further comprising storing at the server version-indicating information corresponding to the differences.

7. The method of claim 1, further comprising merging, by the second device, the differences with third workspace data stored on the second device.

8. A system for synchronizing workspace data, comprising:

means for storing first workspace data on a first device;

means for storing second workspace data on a second device;

means for determining differences between the first workspace data and the second workspace data;

means for storing the differences at a global server; and

means for sending the differences from the global server to the second device.

9. The system of claim 8, wherein the first workspace data comprises a workspace data element from a first user of the first device to a second user of the second device.

10. The system of claim 9, wherein the workspace data element includes data selected from a group including email data, file data, calendar data, user data and bookmark data.

11. The system of claim 8, wherein at least one of the first device and the second device is selected from a group including a smart phone, a television settop box and a personal computer.

12. The system of claim 8, further comprising means for continuing to store the differences at the global server is continued after the sending.

13. The system of claim 8, further comprising means for storing at the server version-indicating information corresponding to the differences.

14. The system of claim 8, further comprising means for merging, by the second device, the differences with third workspace data stored on the second device

* * * * *